Exercise 1: Test your Subversion account:

1. Open the Web browser Safari
2. Open the VSO master directory: http://wwwbruegge.in.tum.de/repos/VSO/
3. Login in with your username and password

Exercise 2: Create your working directory

1. Open the Mac OS X Terminal (Applications/Utilities)
2. Change to the destination directory for your working directory (e.g. Documents)
3. Checkout the VSO master directory

```
lapbruegge85:~/tmp wolft$ svn checkout
http://wwwbruegge.in.tum.de/repos/VSO/ vso
Authentication realm: <http://wwwbruegge.in.tum.de:80> VSO
Password for 'wolft':
A    vso/trunk
A    vso/trunk/cal
A    vso/trunk/cal/VSO.ics
A    vso/trunk/tutorial
A    vso/trunk/tutorial/audio.txt
A    vso/trunk/tutorial/tracking.txt
A    vso/trunk/tutorial/video.txt
A    vso/trunk/tutorial/orchestra.txt
A    vso/trunk/tutorial/ui.txt
A    vso/trunk/tutorial/seminar.txt
A    vso/trunk/dev
A    vso/trunk/dev/audio
:
:
:
Checked out revision 15.
lapbruegge85:~/tmp wolft$
```

Exercise 3: Adding a file to the master directory

1. Start the text editor SubEthaEdit. Located in the directory Applications
2. Create a new text file containing your name and some text of your choice.
3. Save the text file to your to the directory …vso/trunk/tutorial using **your** name. For instance: TimoWolf.txt
4. Switch to the Terminal and change to the tutorial directory.

```
lapbruegge85:~/tmp wolft$ cd vso/trunk/tutorial/
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$
```

5. add the new file to your subversion working directory

```
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$ svn add TimoWolf.txt
A         TimoWolf.txt
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$
```

6. Commit the new file to the subversion master directory and provide a meaningful message.

```
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$ svn commit -m'Added the
text file about Timo Wolf'
Adding         tutorial/TimoWolf.txt
Transmitting file data .
```

```
Committed revision 16.
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$
```

Exercise 4: Update your working directory

1. Open the Mac OS X Terminal and change to the tutorial directory of your VSO working directory
2. Update your local working directory with changes made at the master directory

```
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$ svn update
At revision 16.
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$
```

Exercise 5: Concurrent modifications

The tutorial directory contains text files for each team:

- audio.txt
- orchestra.txt
- seminar.txt
- tracking.txt
- ui.txt
- video.txt

1. Use SubEthaEdit to add some text and your name to the file of your team. The participants of the Seminar should modify the seminar.txt file.
2. Save your changes
3. Use subversion to identify your local changes

```
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$ svn status
M      video.txt
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$
```

4. Use subversion to see the differences of your working directory

```
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$ svn diff
Index: video.txt
===================================================================
--- video.txt   (revision 16)
+++ video.txt   (working copy)
@@ -0,0 +1 @@
+Some test text(Timo Wolf)
\ No newline at end of file
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$
```

5. Commit your changes to the subversion master directory. Use the svn commit, as described in exercise 3

```
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$ svn commit -m'meaningful
message'
Sending        tutorial/video.txt
Transmitting file data .
Committed revision 17.
lapbruegge85:~/tmp/vso/trunk/tutorial wolft$
```

6. Update your local working directory as described in exercise 4
7. Identify the changes of the team members

SVN Commands

(From the online book "*Version Control with* Subversion". See VSO Portal)

Name: svn add

svn add — Add files, directories, or symbolic links.

Synopsis

`svn add PATH...`

Description

Add files, directories, or symbolic links to your working copy and schedule them for addition to the repository. They will be uploaded and added to the repository on your next commit. If you add something and change your mind before committing, you can unschedule the addition using `svn revert`.

Alternate Names

None

Changes

Working Copy

Accesses Repository

No

Switches

```
--targets FILENAME
--non-recursive (-N)
--quiet (-q)
--config-dir DIR
--auto-props
--no-auto-props
--force
```

Name: svn checkout

svn checkout — Check out a working copy from a repository.

Synopsis

`svn checkout URL[@REV]... [PATH]`

Description

Check out a working copy from a repository. If *`PATH`* is omitted, the basename of the URL will be used as the destination. If multiple URLs are given each will be checked out into a subdirectory of PATH, with the name of the subdirectory being the basename of the URL.

Alternate Names

co

Changes

Creates a working copy.

Accesses Repository

Yes

Switches

```
--revision (-r) REV
--quiet (-q)
--non-recursive (-N)
--username USER
--password PASS
--no-auth-cache
--non-interactive
--config-dir DIR
```

Name: svn commit

svn commit — Send changes from your working copy to the repository.

Synopsis

`svn commit [PATH...]`

Description

Send changes from your working copy to the repository. If you do not supply a log message with your commit by using either the `--file` or `--message` switch, `svn` will launch your editor for you to compose a commit message. See the `editor-cmd` section in the section called "Config".

Tip

If you begin a commit and Subversion launches your editor to compose the commit message, you can still abort without committing your changes. If you want to cancel your commit, just quit your editor without saving your commit message and Subversion will prompt you to either abort the commit, continue with no message, or edit the message again.

Alternate Names

ci (short for "check in"; not "co", which is short for "checkout")

Changes

Working copy, repository
Accesses Repository
Yes
Switches
```
--message (-m) TEXT
--file (-F) FILE
--quiet (-q)
--non-recursive (-N)
--targets FILENAME
--force-log
--username USER
--password PASS
--no-auth-cache
--non-interactive
--encoding ENC
--config-dir DIR
```

Name: svn diff

svn diff — Display the differences between two paths.
Synopsis
```
diff [-r N[:M]] [TARGET[@REV]...]
diff [-r N[:M]] --old OLD-TGT[@OLDREV] [--new NEW-TGT[@NEWREV]] [PATH...]
diff OLD-URL[@OLDREV] NEW-URL[@NEWREV]
```
Description
Display the differences between two paths. The three different ways you can use `svn diff` are:
`svn diff [-r N[:M]] [--old OLD-TGT] [--new NEW-TGT] [PATH...]` displays the differences between *`OLD-TGT`* and *`NEW-TGT`*. If *`PATH`*s are given, they are treated as relative to *`OLD-TGT`* and *`NEW-TGT`* and the output is restricted to differences in only those paths. *`OLD-TGT`* and *`NEW-TGT`* may be working copy paths or *`URL`*`[@`*`REV`*`]`. *`OLD-TGT`* defaults to the current working directory and *`NEW-TGT`* defaults to *`OLD-TGT`*. *`N`* defaults to `BASE` or, if *`OLD-TGT`* is a URL, to `HEAD`. *`M`* defaults to the current working version or, if *`NEW-TGT`* is a URL, to `HEAD`. `svn diff -r N` sets the revision of *`OLD-TGT`* to *`N`*, `svn diff -r N:M` also sets the revision of *`NEW-TGT`* to *`M`*.
`svn diff -r N:M URL` is shorthand for `svn diff -r N:M --old=URL --new=URL`.
`svn diff [-r N[:M]] URL1[@N] URL2[@M]` is shorthand for `svn diff [-r N[:M]] --old=URL1 --new=URL2`.
If *`TARGET`* is a URL, then revs N and M can be given either via the `--revision` or by using “@” notation as described earlier.
If *`TARGET`* is a working copy path, then the `--revision` switch means:
`--revision N:M`
The server compares *`TARGET`*@*`N`* and *`TARGET`*@*`M`*.
`--revision N`
The client compares *`TARGET`*@*`N`* against working copy.
(no `--revision`)
The client compares base and working copies of *`TARGET`*.
If the alternate syntax is used, the server compares *`URL1`* and *`URL2`* at revisions *`N`* and *`M`* respectively. If either *`N`* or *`M`* are omitted, a value of `HEAD` is assumed.
By default, `svn diff` ignores the ancestry of files and merely compares the contents of the two files being compared. If you use `--notice-ancestry`, the ancestry of the paths in question will be taken into consideration when comparing revisions (that is, if you run `svn diff` on two files with identical contents but different ancestry you will see the entire contents of the file as having been removed and added again).
Alternate Names
di
Changes
Nothing
Accesses Repository
For obtaining differences against anything but `BASE` revision in your working copy
Switches
```
--revision (-r) REV
--old OLD-TARGET
--new NEW-TARGET
--extensions (-x) "ARGS"
--non-recursive (-N)
--diff-cmd CMD
--notice-ancestry
--username USER
--password PASS
--no-auth-cache
--non-interactive
--no-diff-deleted
--config-dir DIR
```

Name: svn status

svn status — Print the status of working copy files and directories.

Synopsis

`svn status [PATH...]`

Description

Print the status of working copy files and directories. With no arguments, it prints only locally modified items (no repository access). With `--show-updates`, add working revision and server out-of-date information. With `--verbose`, print full revision information on every item.

The first five columns in the output are each one character wide, and each column gives you information about different aspects of each working copy item.

The first column indicates that an item was added, deleted, or otherwise changed.

' '
No modifications.
'A'
Item is scheduled for Addition.
'D'
Item is scheduled for Deletion.
'M'
Item has been modified.
'R'
Item has been replaced in your working copy.
'C'
Item is in conflict with updates received from the repository.
'X'
Item is related to an externals definition.
'I'
Item is being ignored (e.g. with the `svn:ignore` property).
'?'
Item is not under version control.
'!'
Item is missing (e.g. you moved or deleted it without using `svn`). This also indicates that a directory is incomplete (a checkout or update was interrupted).
'~'
Item is versioned as one kind of object (file, directory, link), but has been replaced by different kind of object.

The second column tells the status of a file's or directory's properties.

' '
No modifications.
'M'
Properties for this item have been modified.
'C'
Properties for this item are in conflict with property updates received from the repository.

The third column is populated only if the working copy directory is locked.

' '
Item is not locked.
'L'
Item is locked.

The fourth column is populated only if the item is scheduled for addition-with-history.

' '
No history scheduled with commit.
'+'
History scheduled with commit.

The fifth column is populated only if the item is switched relative to its parent (see the section called “Switching a Working Copy”).

' '
Item is a child of its parent directory.
'S'
Item is switched.

The out-of-date information appears in the eighth column (only if you pass the `--show-updates` switch).

' '
The item in your working copy is up-to-date.
'*'
A newer revision of the item exists on the server.

The remaining fields are variable width and delimited by spaces. The working revision is the next field if the `--show-updates` or `--verbose` switches are passed.

If the `--verbose` switch is passed, the last committed revision and last committed author are displayed next.

The working copy path is always the final field, so it can include spaces.

Alternate Names

stat, st

Changes

Nothing

Accesses Repository

Only if using `--show-updates`

Switches

`--show-updates (-u)`

```
--verbose (-v)
--non-recursive (-N)
--quiet (-q)
--no-ignore
--username USER
--password PASS
--no-auth-cache
--non-interactive
--config-dir
```

Name svn update

svn update — Update your working copy.
Synopsis
```
svn update [PATH...]
```
Description
`svn update` brings changes from the repository into your working copy. If no revision given, it brings your working copy up-to-date with the `HEAD` revision. Otherwise, it synchronizes the working copy to the revision given by the `--revision` switch.
For each updated item a line will start with a character reporting the action taken. These characters have the following meaning:
A
Added
D
Deleted
U
Updated
C
Conflict
G
Merged
A character in the first column signifies an update to the actual file, while updates to the file's properties are shown in the second column.
Alternate Names
up
Changes
Working copy
Accesses Repository
Yes
Switches
```
--revision (-r) REV
--non-recursive (-N)
--quiet (-q)
--diff3-cmd CMD
--username USER
--password PASS
--no-auth-cache
--non-interactive
--config-dir DIR
```