



DM

# CASTOR

## from CVS to SVN

- Why to move to SVN
- Almost no change to daily life
- Nevertheless
  - Conceptual differences
  - Slight command variations
  - New commands
  - New tools
- The conversion process

- CVS is extremely old
  - Less powerful than current tools
    - e.g. : no moves, no merges
  - Soon (1 year) unmaintained at CERN
- GIT or mercurial are felt too complex ( understand powerful ? )
  - Big gap compared to CVS
  - Will not be supported by CERN
- SVN is the right compromise
  - Supported by CERN

|          | CVS           | SVN          |
|----------|---------------|--------------|
| checkout | cvcs checkout | svn checkout |
| update   | cvcs update   | svn update   |
| diff     | cvcs diff     | svn diff     |
| commit   | cvcs commit   | svn commit   |

And so on for svn add, delete, log, annotate (aka blame), import ...

- I have 11 more slides coming detailing the differences :-)
- Differences can be
  - conceptual
  - slight changes of syntax in commands
  - new features you want to take benefit from (e.g. move, merge)
  - new tools (svnweb, trac, mails)
- So let's go through all this...

- No more ',v' files, but a DB
    - So no direct, hacky access
  - Hierarchical structure of the repository:
    - trunk
    - branches
      - v2\_1\_8Version
    - Tags
      - v2\_1\_8\_10
    - Imakefile
    - castor
      - stager
      - ...
    - ns
      - Imakefile
      - ...
      - ...
- Notion of changeset : all files modified/created/dropped/moved within a given commit
  - identified by a unique number (starting with 1 and increasing with time cross branches/tags)
  - Also identified by uuids

- Files/directories can have properties
  - these store extra info on files
    - e.g. mergeinfo, ignore
  - New properties can be arbitrary added by users
    - do not do it without wide agreement
- .cvsignore files are gone
  - replaced by svn:ignore property on the directory
- Take care that you can commit into a tag
  - This should never be done in our context
  - So take care to always checkout branches
    - Or make your tag checkout read only

- Was

```
cvs -d <rep> co -r <branch> -d <dir> <module>
```

- Now

```
svn co <URL> <dir>
```

where URL contains rep, branch and module, using the hierarchical repository approach, e.g. :

```
svn+ssh://svn/repos/CASTOR/CASTOR2/trunk
```

```
svn+ssh://svn/repos/CASTOR/CASTOR2/branches/v2_1_8Version
```

```
svn+ssh://svn/repos/CASTOR/CASTOR2/tags/v2_1_8_10
```

Note by the way that “HEAD” is now named “trunk”

Also take care of not checking out the whole hierarchy !



- commit, update, add and delete do not change
- annotate neither but its primary name is blame ;-)
- diff does not change, except that you need to use URLs, as in the checkout command or revision numbers

```
svn diff svn+ssh://svn/repos/CASTOR/CASTOR2/trunk  
svn diff -r 10306
```

However, the default (no argument) is still to diff with the latest version of the current branch.

- In case an update creates a conflict, the behavior is slightly different from CVS
  - SVN will create locally 4 files for you : the attempted merge (like CVS), but also the 2 clashing versions and their common ancestor
  - You need to solve the diff by hand as before, but you also need to tell svn via the `svn resolved` command

- `svn switch` allows you to change a checkout repository from one branch to another, keeping the local modifications. Nice if you worked on wrong branch
- `svn move / rename` and `svn copy` allow to move/rename/copy a file within the repository.
  - And no history is lost :-)
- `svn merge` allows to automatically merge a given changeset into the current branch
  - `svn` will then know that this is a backport
  - example of backporting commit 10306 to a branch :  

```
svn merge svn+ssh://svn/repos/CASTOR/CASTOR2/trunk@16305  
svn+ssh://svn/repos/CASTOR/CASTOR2/trunk@10306
```
  - This should now be the only way used to backport

- `svn proplist` lists properties of a file
- `svn propget`, `svn propset` retrieve and set the value of a given property
- `svn propdel` drops a property
- `svn propedit` allows to edit a property inside your favorite editor (useful for `svn:ignore`)
- Note that properties are versionned and committed like the content of the file and that you can have property merging conflicts :-)
  - You will have a `.prej` file created in such a case detailing the situation. Fix and use `svn resolved`

- `svn revert` can be used to revert all your modifications (including all changes to properties)
- `svn info` tells you everything about your file (revision, URL of the repository, last change, checksum, etc...)
- `svn status` tells you the status of your files without having to do an `svn update` and without going to the server
- Finally `svn help` is your best friend :-)

- WebSVN is a tool to browse an SVN repository  
<http://svnweb.cern.ch/world/wsvn/CASTOR>
- Trac is another tool to browse and it has a ticket system (please do not use it) plus a revision tree viewer  
<https://svnweb.cern.ch/trac/CASTOR/browser>
- Mails will still be sent on each commit, with links to both WebSVN and trac diff pages
  - On top, short diffs will now be inlined

- It uses the cvs2svn tool + some home made scripts using the svndump tool python interface
- It tries to create a clean SVN repository
  - old modules were dropped (SRM1, CASTOR, PROTO2)
  - old branches and tags were dropped (till 2.1.6\* inc)
  - old upgrades are dropped (till 2.1.6\* inc)
  - RH.xmi was cleaned up (till 2.1.6\* inc)
  - .cvsignore are converted into ignore attributes
  - multi files commit are reassembled into changesets
  - backports are converted to merges
    - as long as they could be recognized as such
    - based on date and commit message
- The AFS checkout will disappear

- Steps are the following
    - CVS will be made read only
      - And will stay forever like this
    - The conversion will be done
    - SVN will be open
  - You have then to
    - Check out CASTOR from SVN
    - move manually your non committed modifications
  - An SVN repository for CASTOR is in place for tests
    - standard conversion process done few weeks ago
    - Fully functional (inc. commits, mails, web, etc...)
      - But modification done there will be erased
- ! PLEASE TEST IT BEFORE THE MOVE !**

- IT svn service : <http://svn.web.cern.ch/svn/index.php>
  - Especially the HOWTO
    - <http://svn.web.cern.ch/svn/howto.php>
- SVN : <http://svnbook.red-bean.com/en/1.5>
  - Especially “SVN for CVS users” part
    - <http://svnbook.red-bean.com/en/1.5/apa.html>
  - And “Basic Work Cycle”
    - <http://svnbook.red-bean.com/en/1.5/ch03s05.html>
- Web browsers :
  - <http://svnweb.cern.ch/world/wsvn/CASTOR>
  - <https://svnweb.cern.ch/trac/CASTOR>
  - Note the difference (http vs https)



- No demo of svn itself, only the web tools
- WebSVN :
  - Go to <http://svnweb.cern.ch/world/wsvn/CASTOR>
  - View the hierarchy under trunk, branches and tags
  - In trunk, click on makesql.sh, see colored browsing
  - Test compare with previous, blame and view log
  - Goto rev 16539 (top left)
  - See multiple files commit (imported from CVS)
  - Select first one and see more complex diff, blame, ...

## Trac :

- Go to <https://svnweb.cern.ch/trac/CASTOR>
- Only “Browse Source” and “Rev Tree” should be used
- Click “Browse Source” and view the hierarchy under trunk, branches and tags
- In trunk, click on makesql.sh, see the file
- Test last change, annotate and revision log
  - See colorization of the annotate depending on age
- Goto rev 16539 (put it in box, go to the file for this revision, then click after the '@' in the path)
  - See multiple files commit and nice diff
- Goto rev 16544 (only change number in URL)
  - See property change due to backport from 16543

## Trac :

- Goto “Rev Tree”
- See graphical view of last month commits and backports
- Especially see the backport from 16543 to 16544 you have just looked at
- Put your mouse on top of revision to get details. Click to see the changeset
- See also the unnamed branch
  - Click on it and see strange changeset due to partial tagging for 2.1.8-8-2
  - Typical example of a commit (here at least properties) on a tag. This kind of creates an unnamed branch