

# Introduction to version control



David Rey – DREAM – 15 dec. 2004

# Overview

- Collaborative work and version control
- Main CVS user commands
- Advanced use of CVS
- CVS server at INRIA Sophia Antipolis



# Overview

- Collaborative work and version control
- Main CVS user commands
- Advanced use of CVS
- CVS server at INRIA Sophia Antipolis



# Collaborative work and version control: examples

- . Development
  - Source files: C, C++, java, Fortran, Tcl, Python, shell scripts, ...
  - Build/config files: Makefile, ant, ...
- . Text documents/articles/bibliographies
  - Plain text
  - Latex/bibtex
- . Web pages
  - Html
  - Php, javascripts, ...
- . XML documents
- . ...

→ ASCII documents



Sophia  
Antipolis

# A software development process at INRIA

- INRIA recommandations about software development:
  - <http://devel.inria.fr/recom/>
  - <http://devel.inria.fr/>
- In particular, concerning version control:
  - <http://devel.inria.fr/recom/processus/main005.html#toc8>



# Version control main ideas

- Distributed documents/collaborative work
  - Automatic merging
  - Alarms on conflicts
  - Ease the communication between users (log messages, emails, ...)
- Version control: incremental versions
  - All previous versions available
  - Minimal necessary disk space (incremental)
  - History of changes/logs



# Version control software

- CVS: <http://www.cvshome.org>
- Arch: <http://www.gnu.org/software/gnu-arch/>
- Subversion: <http://subversion.tigris.org/>
- Forges that use version control software:  
([http://devel.inria.fr/recom/processus/main005.html#developpement\\_forges](http://devel.inria.fr/recom/processus/main005.html#developpement_forges))
  - GForge : [www.gforge.org](http://www.gforge.org)
  - Savannah : [savannah.gnu.org](http://savannah.gnu.org)
  - LibreSource : [libresource.inria.fr](http://libresource.inria.fr)
  - Visual Source Safe (pour Windows) : [msdn.microsoft.com/vstudio/previous/ssafe](http://msdn.microsoft.com/vstudio/previous/ssafe)



# CVS: Concurrent Version System

- Widely used on a lot of different platforms (Linux, Windows, ...)
- Open source license
- For local and/or distant usage
- Recommended for INRIA software developments
- Several clients possible
  - Command line
  - GUI: tkCVS, jCVS, WebCvs, Jalindi igloo, WinCVS, TortoiseCVS





# What CVS is for ?

- Several users work together at the same time on the same files (concurrency)
- Version control
  - Tags
  - Version comparisons
  - Multiple lines of development in the same base
  - Branching
  - Support for binary files
  - Event control (e.g. notification)



# What CVS is not for ?

- Backup
- Bug tracking
- Source documentation
- Dependencies
- ...



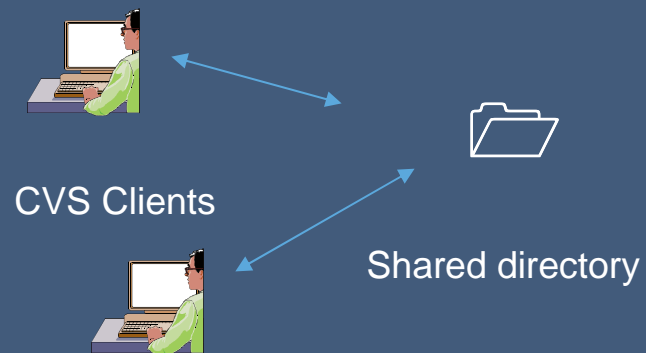
# Overview

- Collaborative work and version control
- **Main CVS user commands**
- Advanced use of CVS
- CVS server at INRIA Sophia Antipolis

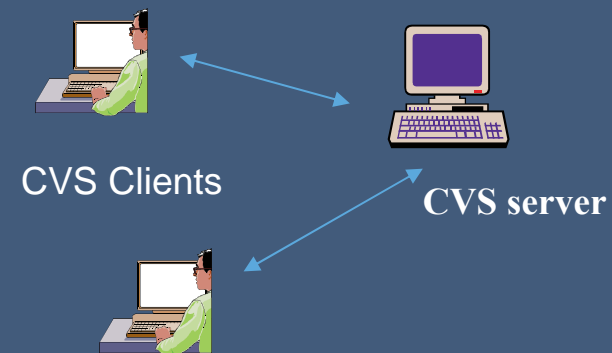


# CVS: client/server architecture

2 possibilities:



Local



Client/Server



Sophia  
Antipolis

# CVS hierarchy

- Repository root or “CVSROOT”
  - Looks like “cvs-sop.inria.fr:/CVS/SmartTools”
  - set the CVSROOT environment variable on UNIX systems
  - configure an item in the setup menus of a GUI client
  - Use the command line: `cvs -d CVSROOT`
- Modules
  - For example SmartTools uses 3 modules: SmartTools, SmartToolsAddons, Documents
- Directories
  - Each module contains an arbitrary hierarchy of directories
- Files (incremental diff versions of)

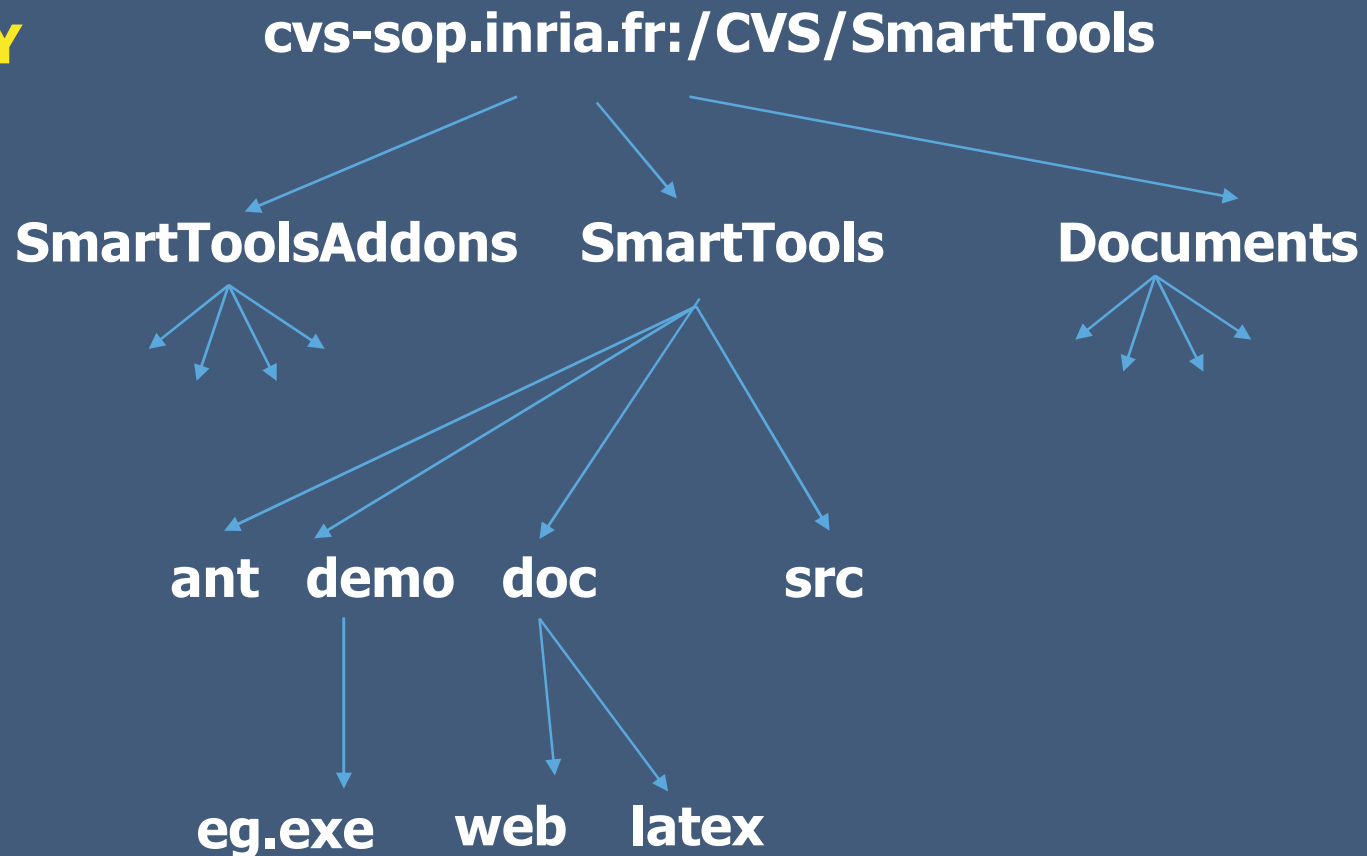


# CVS hierarchy : an example

**REPOSITORY  
(CVSROOT)**

**MODULES**

**FOLDERS  
AND FILES**



Sophia  
Antipolis

# CVS help

- `cv`s --help-commands (list available commands)
- `cv`s --help-options (list general options)
- `cv`s -H import (specific help for a command and its specific options, e.g. “import”)

→ CVS general options + specific command options  
are different :

`cv`s [general options] <command> [specific options]





mythos-developer \$ cvs --help-commands

CVS commands are:

add	Add a new file/directory to the repository
admin	Administration front end for rcs
annotate	Show last revision where each line was modified
checkout	Checkout sources for editing
commit	Check files into the repository
diff	Show differences between revisions
edit	Get ready to edit a watched file
editors	See who is editing a watched file
export	Export sources from CVS, similar to checkout
history	Show repository access history
import	Import sources into CVS, using vendor branches
init	Create a CVS repository if it doesn't exist
kserver	Kerberos server mode
log	Print out history information for files
login	Prompt for password for authenticating server
logout	Removes entry in .cvspass for remote repository
pserver	Password server mode
rannotate	Show last revision where each line of module was modified
rdiff	Create 'patch' format diffs between releases
release	Indicate that a Module is no longer in use
remove	Remove an entry from the repository
rlog	Print out history information for a module
rtag	Add a symbolic tag to a module
server	Server mode
status	Display status information on checked out files
tag	Add a symbolic tag to checked out version of files
unedit	Undo an edit command
update	Bring work tree in sync with repository
version	Show current CVS version(s)
watch	Set watches
watchers	See who is watching a file

(Specify the --help option for a list of other help options)

mythos-developer \$



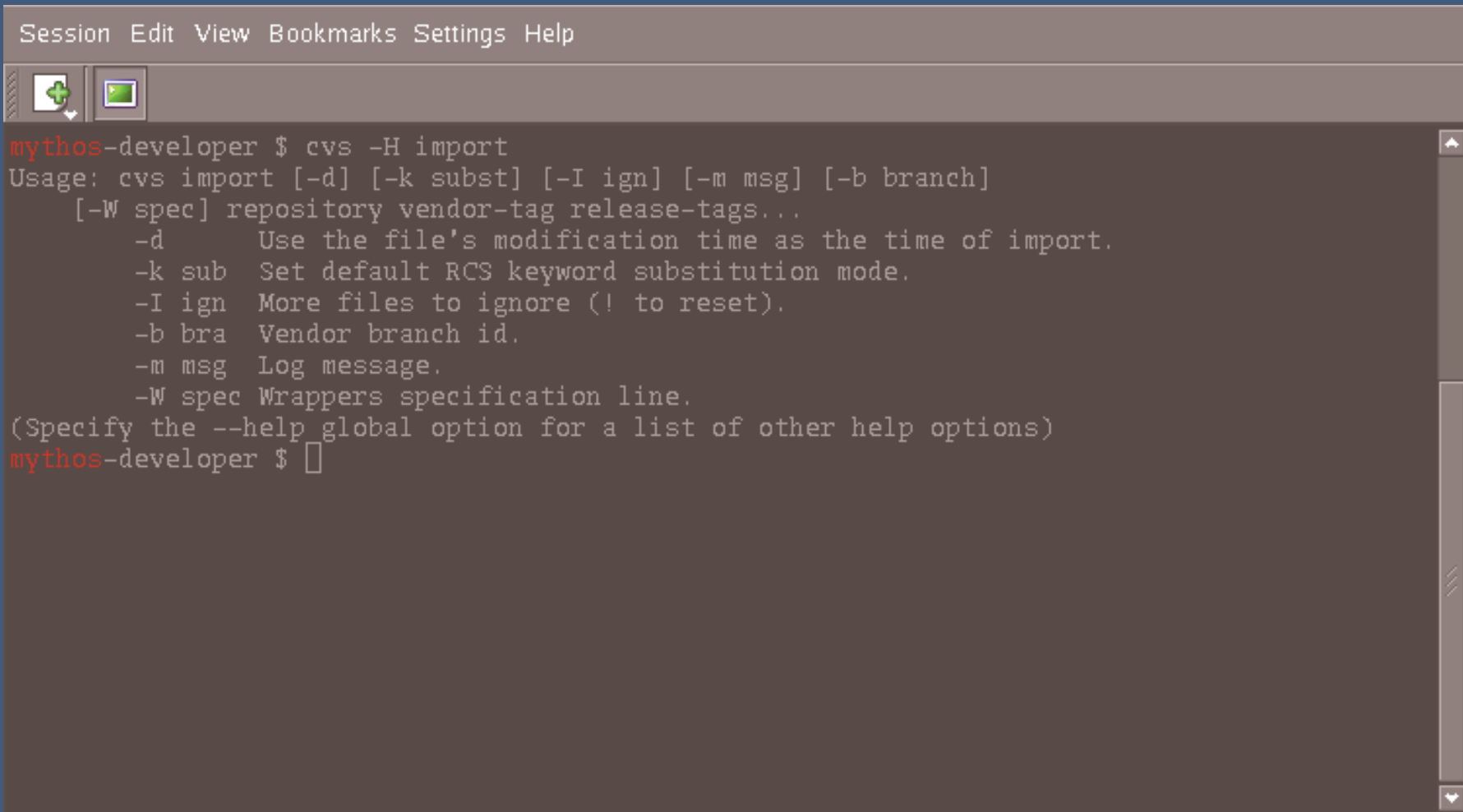
# CVS --help-options

```
Session Edit View Bookmarks Settings Help

mythos-developer $ cvs --help-options
CVS global options (specified before the command name) are:
  -H          Displays usage information for command.
  -Q          Cause CVS to be really quiet.
  -q          Cause CVS to be somewhat quiet.
  -r          Make checked-out files read-only.
  -w          Make checked-out files read-write (default).
  -n          Do not execute anything that will change the disk.
  -t          Show trace of program execution -- try with -n.
  -v          CVS version and copyright.
  -T tmpdir   Use 'tmpdir' for temporary files.
  -e editor   Use 'editor' for editing log information.
  -d CVS_root Overrides $CVSROOT as the root of the CVS tree.
  -f          Do not use the ~/.cvsrc file.
  -z #        Use compression level '#' for net traffic.
  -x          Encrypt all net traffic.
  -a          Authenticate all net traffic.
  -s VAR=VAL  Set CVS user variable.
(Specify the --help option for a list of other help options)
mythos-developer $
```



# CVS -H import



The screenshot shows a terminal window with a menu bar (Session, Edit, View, Bookmarks, Settings, Help) and a toolbar with a green plus icon and a green square icon. The terminal text is as follows:

```
mythos-developer $ cvs -H import
Usage: cvs import [-d] [-k subst] [-I ign] [-m msg] [-b branch]
      [-W spec] repository vendor-tag release-tags...
      -d      Use the file's modification time as the time of import.
      -k sub   Set default RCS keyword substitution mode.
      -I ign   More files to ignore (! to reset).
      -b bra   Vendor branch id.
      -m msg   Log message.
      -W spec  Wrappers specification line.
(Specify the --help global option for a list of other help options)
mythos-developer $
```



# Import

- Used to synchronize a local tree as a whole with the server
  - Most often used to initialize the server tree
  - Can also be used after this initial step at any time (to avoid one-by-one files addition)
- 
- `cd local_directory`
  - `cvs import -m "message" directory_on_base branch_name version_name`



# Checkout and update

- `cv`s -d cvsroot checkout module\_name
- `cv`s update -d directories/files
  - U: updated
  - P: the version on the server is newer and the local copy has been updated
  - M: modified (or merged), the local file is different from the server file and merge was done in the local copy
  - ?: unknown, this file is present locally and not in the CVS base
  - A: added, this file is present locally and has been added (before commit)
  - R: removed, this file is not present locally and has been removed (before commit)
  - C: conflict, the local file has been marked with conflict markers which identify the different conflict locations





```
mythos-test $ cvs -d :ext:drey@cvs-sop.inria.fr:/CVS/SmartTools checkout Documents
cvs checkout: Updating Documents
U Documents/CSSinSmartTools.html
U Documents/Composants.html
U Documents/Composants.txt
U Documents/DidierMemoST
U Documents/Event.txt
U Documents/MissionDream2004.txt
U Documents/SmartTools.ppt
U Documents/compilation.txt
U Documents/packages.html
U Documents/packages.txt
cvs checkout: Updating Documents/PawelKowalski
U Documents/PawelKowalski/changements.txt
U Documents/PawelKowalski/propositions.txt
U Documents/PawelKowalski/rapport.txt
U Documents/PawelKowalski/soapmon.txt
U Documents/PawelKowalski/tests.txt
U Documents/PawelKowalski/todos.txt
cvs checkout: Updating Documents/PawelKowalski/TransformTutorial
U Documents/PawelKowalski/TransformTutorial/TODOS.txt
U Documents/PawelKowalski/TransformTutorial/TransformTutorial.tcp
U Documents/PawelKowalski/TransformTutorial/TransformTutorial.tex
U Documents/PawelKowalski/TransformTutorial/biblio.tex
U Documents/PawelKowalski/TransformTutorial/createCoXSL.tex
U Documents/PawelKowalski/TransformTutorial/generate-component.vsd
U Documents/PawelKowalski/TransformTutorial/generate-wslayer.vsd
U Documents/PawelKowalski/TransformTutorial/generateWsLayer.tex
U Documents/PawelKowalski/TransformTutorial/graphDeDepart.dot
U Documents/PawelKowalski/TransformTutorial/intro.tex
U Documents/PawelKowalski/TransformTutorial/output-dir.dot
U Documents/PawelKowalski/TransformTutorial/overview.vsd
U Documents/PawelKowalski/TransformTutorial/ressource-dir.dot
U Documents/PawelKowalski/TransformTutorial/src-directory.dot
U Documents/PawelKowalski/TransformTutorial/subdivision.tex
U Documents/PawelKowalski/TransformTutorial/tutorial.tex
U Documents/PawelKowalski/TransformTutorial/tutorial.txt
```

Session Edit View Bookmarks Settings Help



```
mythos-developer $ cvs update -d .
? src/fnf.class
? src/ftpWwwZips.class
? src/stgen.class
? src/zipnf.class
cvs update: Updating .
M build.xml
A newfile
cvs update: warning: smarttools.policy was lost
U smarttools.policy
cvs update: Updating src
mythos-developer $
```



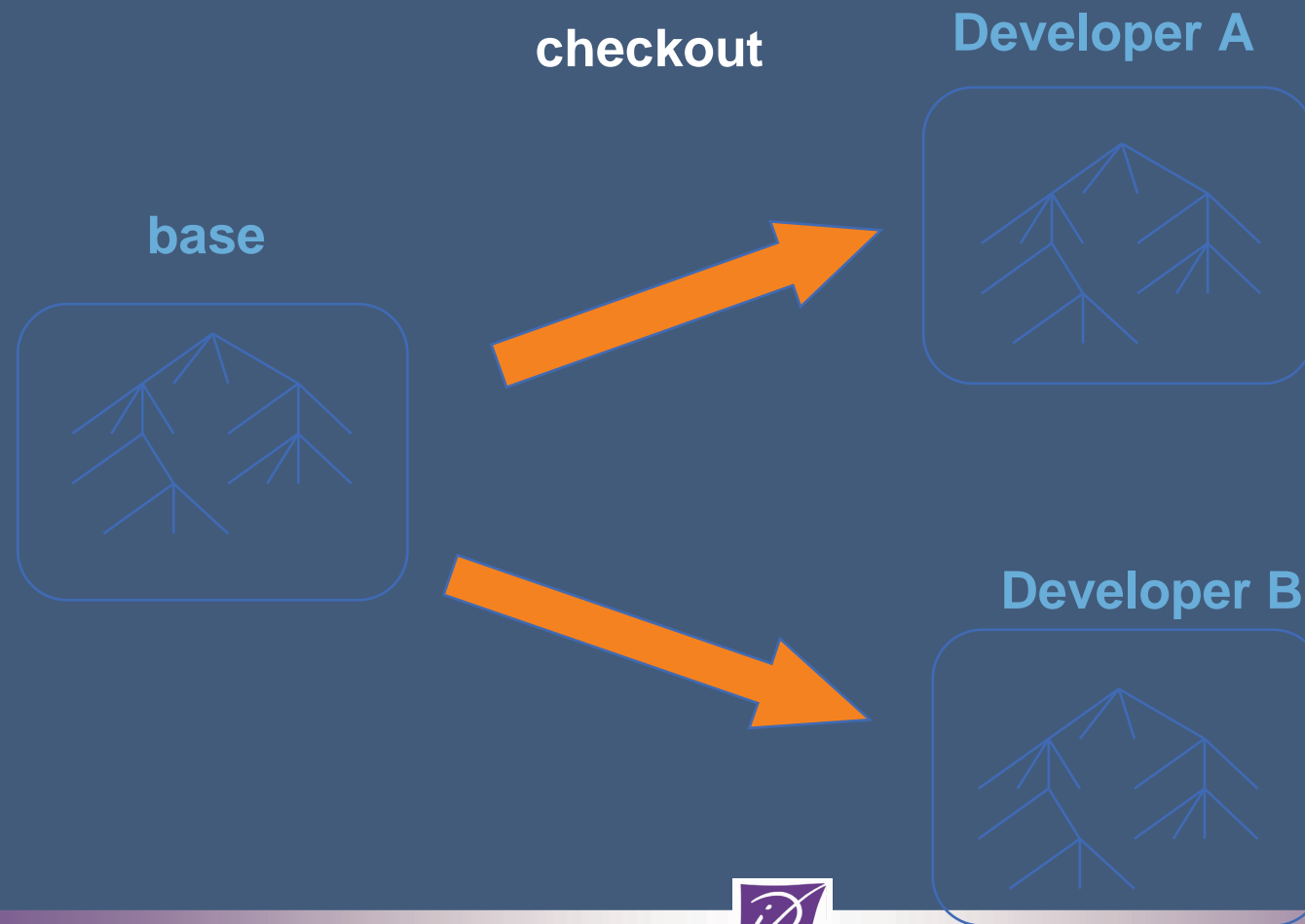
Sophia  
Antipolis

# Commit

- `cvcs commit -m "my message" directories/files`
- Always use explicit messages !!!
- List of forbidden messages:
  - "my message" ☺
  - "" → ?
  - "modification of file toto.tex" → which modification ?
  - "bug fixed" → which bug ?
  - ...



# Ideal Development (1/4)





# Ideal Development (2/4)

development

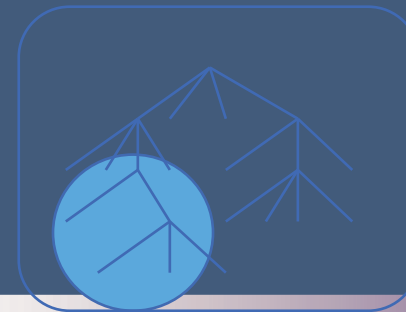
Developer A



base

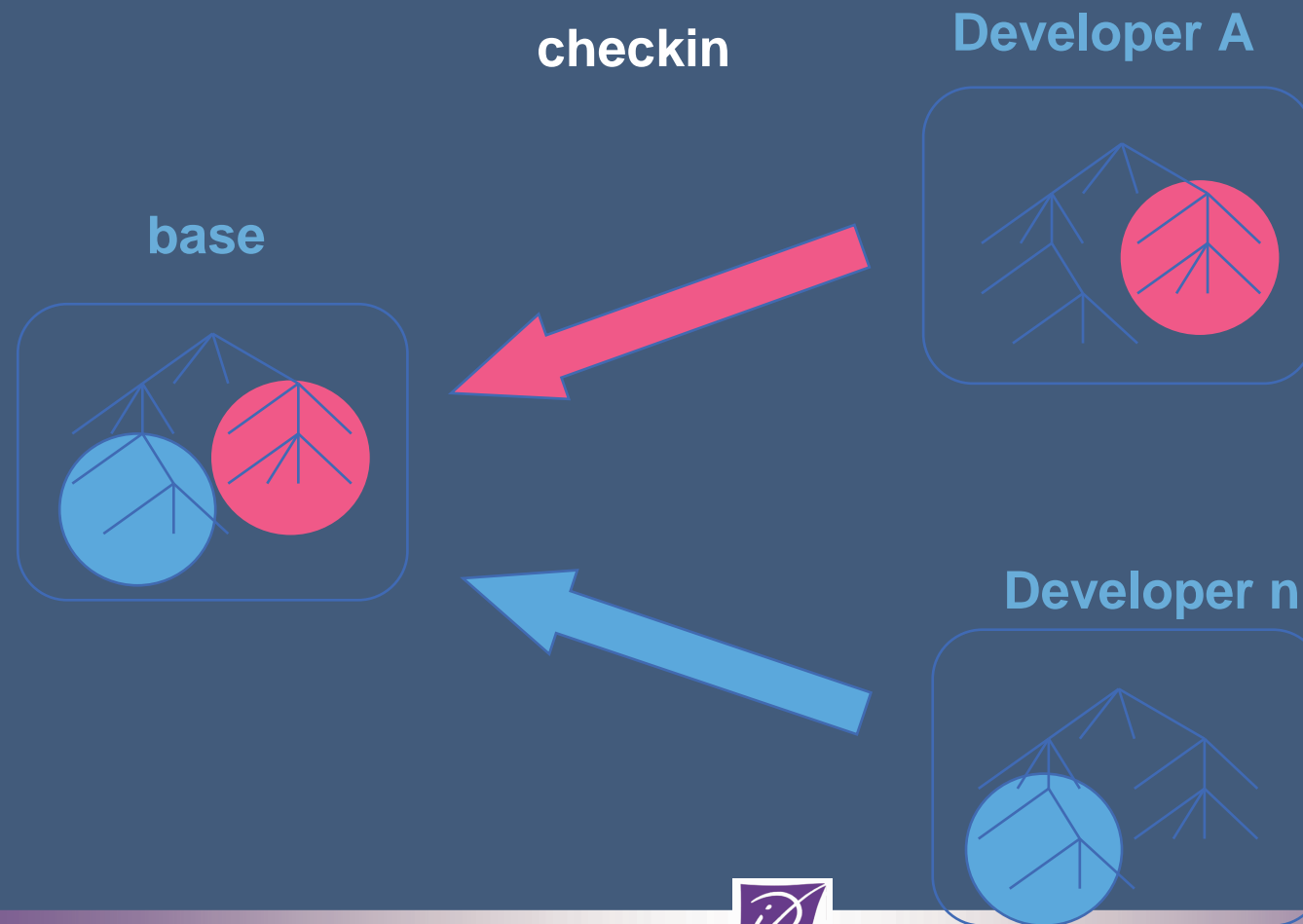


Developer B

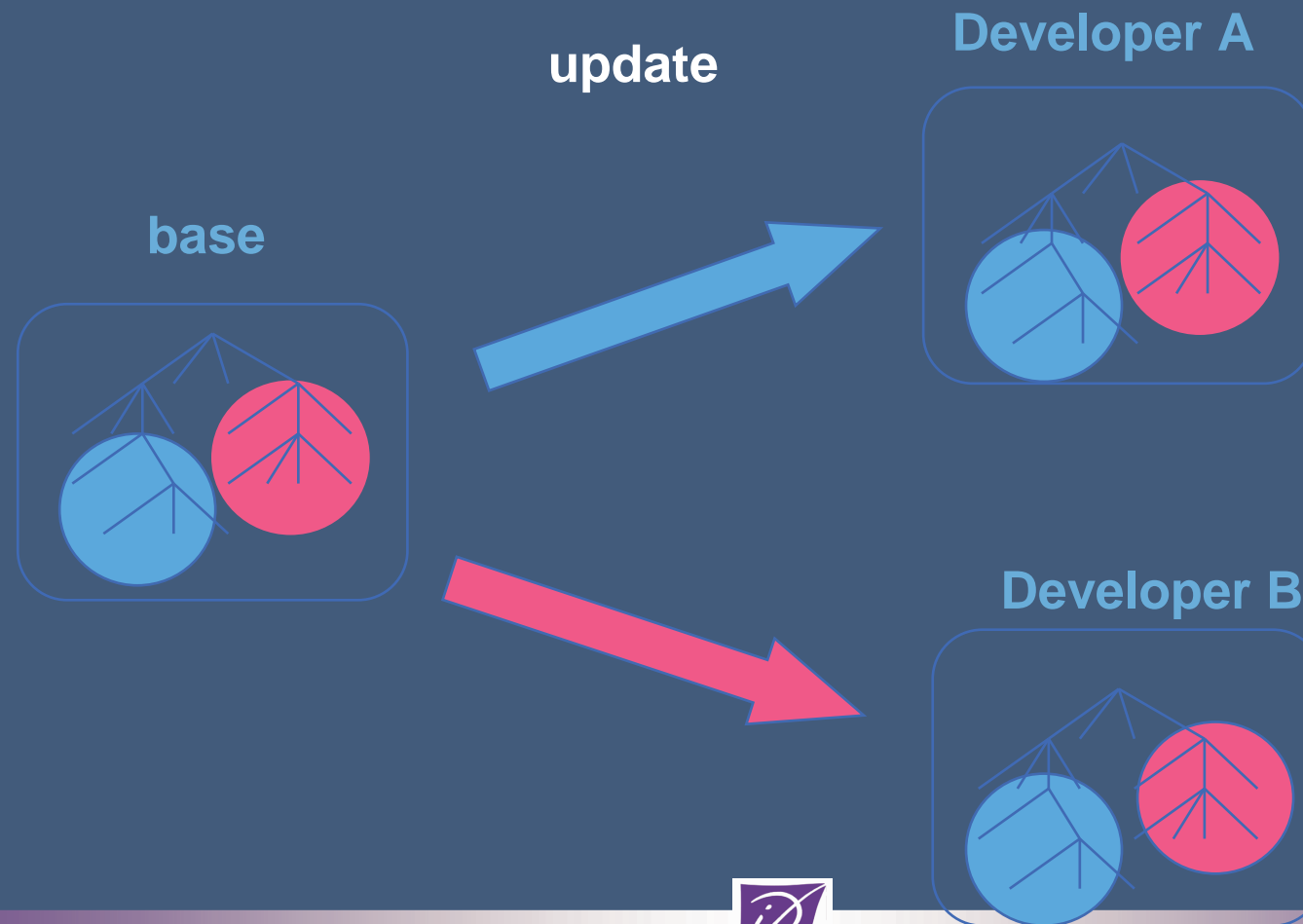


Sophia  
Antipolis

# Ideal Development (3/4)



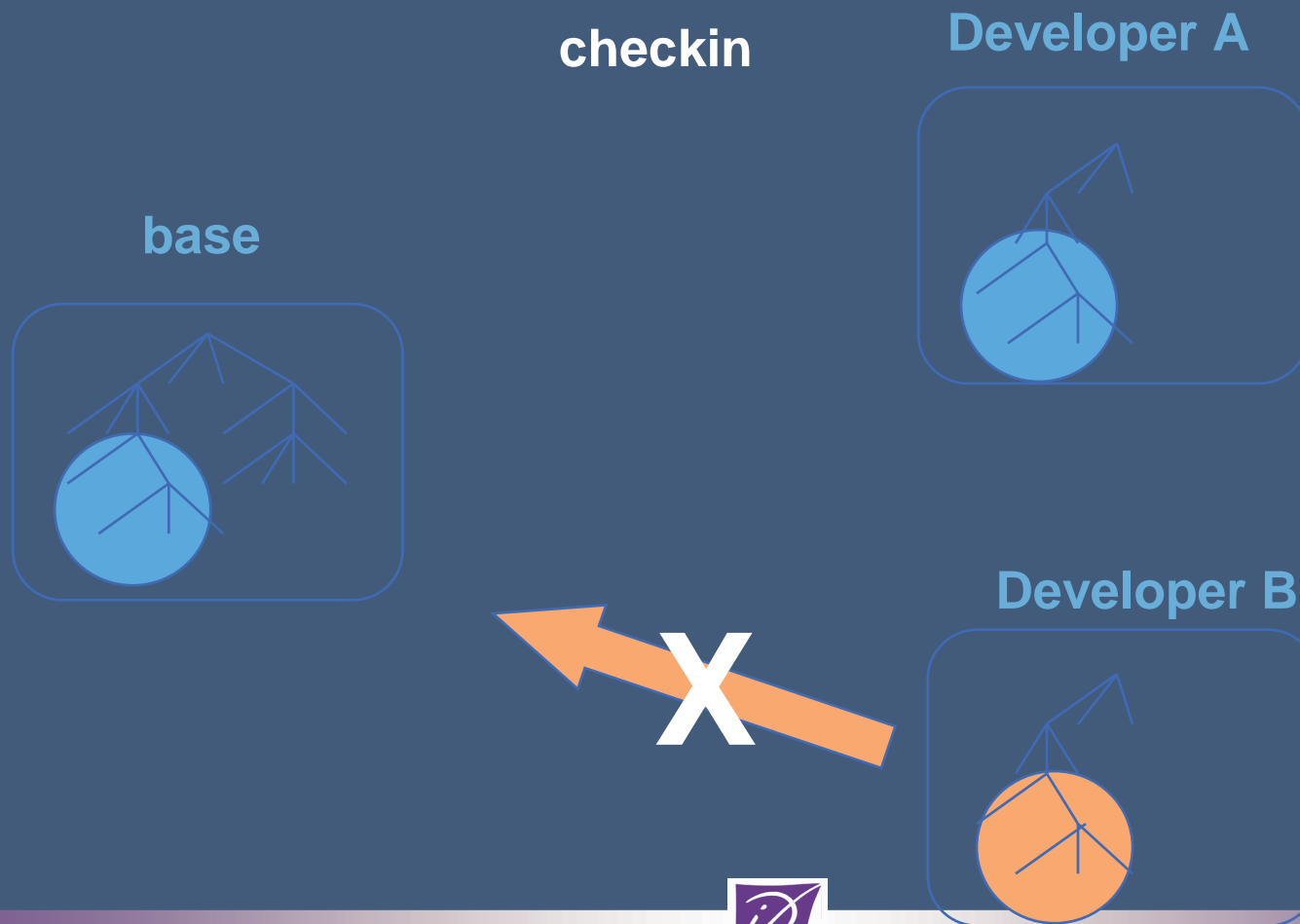
# Ideal Development (4/4)



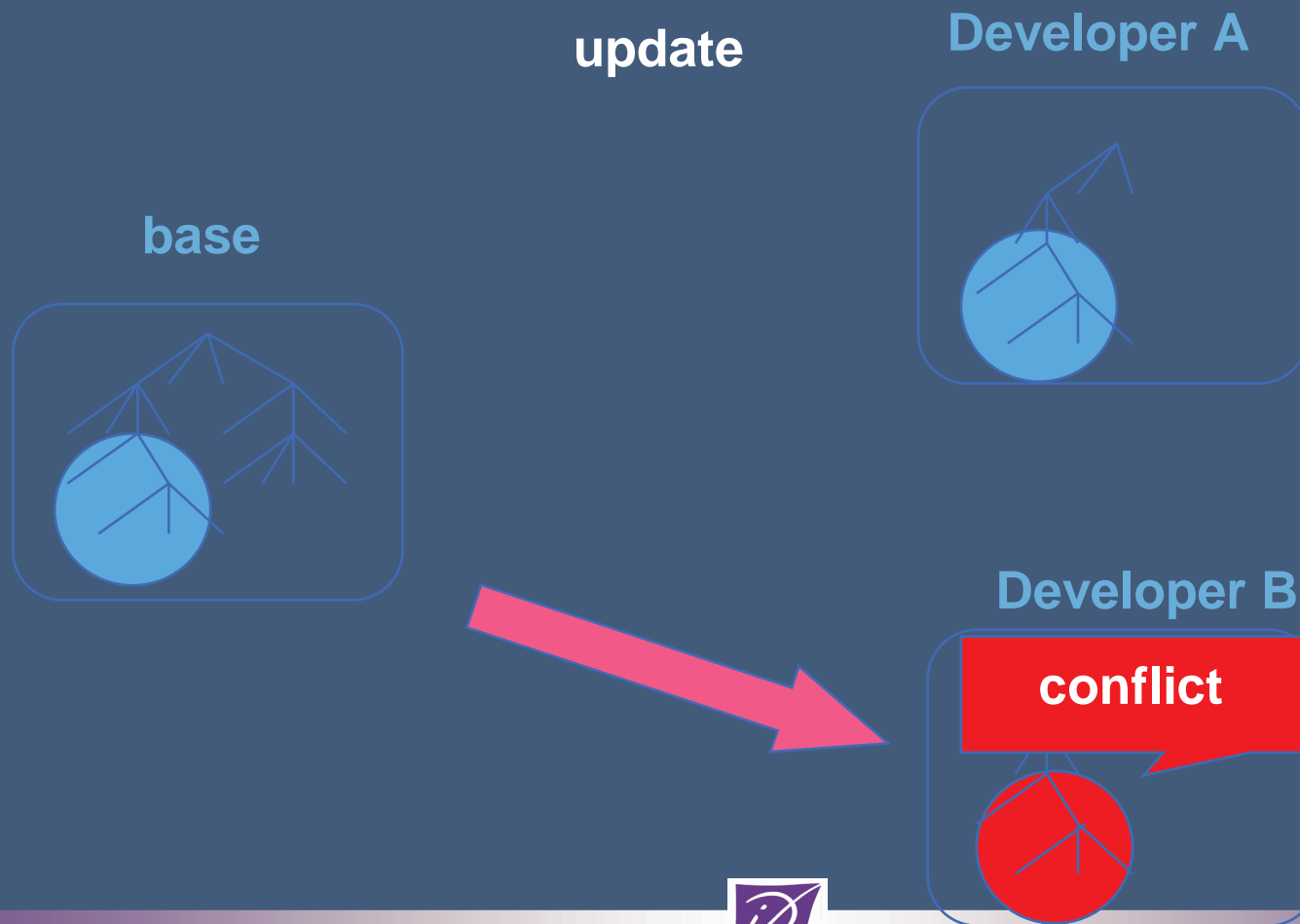
# Real Development (1/5)



# Real Development (2/5)



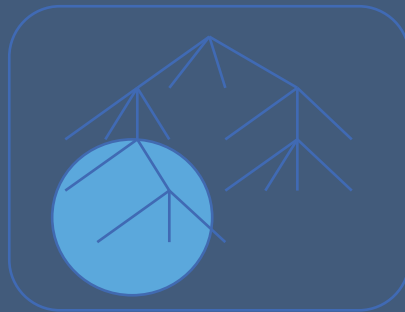
# Real Development (3/5)



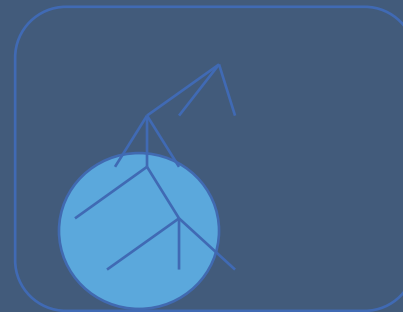
# Real Development (4/5)

## Conflict Resolution

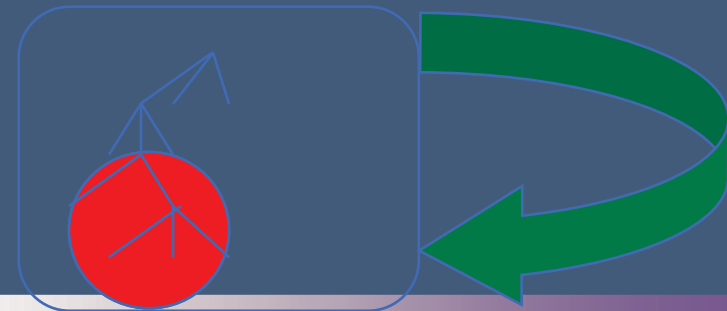
base



## Developer A

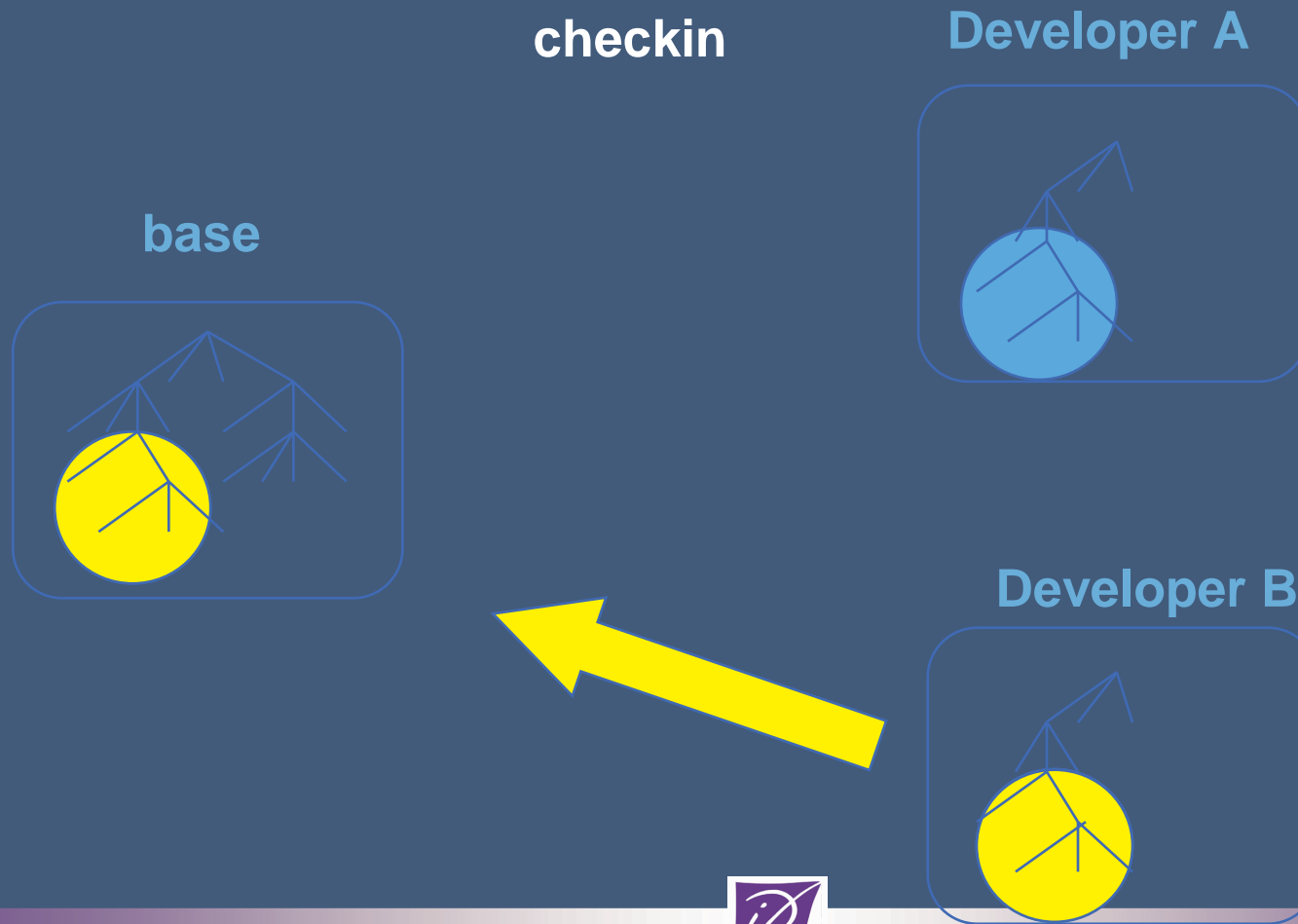


## Developer B



Sophia  
Antipolis

# Real Development (5/5)





# Conflicts

- Do not panic!
  - Many problems are easy to fix
  - Generally, it does not require a lot of interaction
- If necessary, discuss with people to find a compromise
  - Meetings
  - Email
  - Phone
  - ...
- Usually can be avoided with regular updates



# Status, log, diff, annotate

- . `cv status filename`
  - Gives information about local file with comparison to base file
- . `cv log filename`
  - Show all the previous versions number, comitter, date and number of lines modified
- . `cv diff filename`
  - Show lines where there are differences between local and base file
- . `cv annotate filename`
  - Gives information line by line: version of introduction, who, when



# Status examples

```
Session Edit View Bookmarks Settings Help
+ 
mythos-developer $ cvs status build.xml
=====
File: build.xml          Status: Locally Modified

Working revision:      1.71
Repository revision: 1.71  /CVS/SmartTools/SmartTools/ant/developer/build.xml,v
Sticky Tag:           (none)
Sticky Date:          (none)
Sticky Options:       (none)

mythos-developer $
```





```
mythos-dream-web-sources $ cvs log seminaires.html.in | more
```

```
RCS file: /CVS/dream/dream-web/seminaires.html.in,v
```

```
Working file: seminaires.html.in
```

```
head: 1.14
```

```
branch:
```

```
locks: strict
```

```
access list:
```

```
symbolic names:
```

```
    version1: 1.1.1.1
```

```
    dream-web: 1.1.1
```

```
keyword substitution: kv
```

```
total revisions: 15;    selected revisions: 15
```

```
description:
```

```
-----
```

```
revision 1.14
```

```
date: 2004/12/14 17:43:23; author: drey; state: Exp; lines: +1 -1
```

```
add a draft version of the cvs presentation for the seminar of the 15 dec 2004
```

```
-----
```

```
revision 1.13
```

```
date: 2004/12/14 15:55:25; author: jmi; state: Exp; lines: +5 -0
```

```
Updates to add the refactoring seminaires and fixes to the text on  
the links page.
```

```
-----
```

```
revision 1.12
```

```
date: 2004/11/24 19:25:34; author: drey; state: Exp; lines: +6 -6
```

```
maj des evenements et des projets drema et ia-odl
```

```
-----
```

```
revision 1.11
```

```
date: 2004/11/16 14:08:44; author: dgeld; state: Exp; lines: +1 -1
```

```
--Encore--
```

Session Edit View Bookmarks Settings Help



```
mythos-developer $ cvs diff build.xml
Index: build.xml
=====
RCS file: /CVS/SmartTools/SmartTools/ant/developer/build.xml,v
retrieving revision 1.71
diff -r1.71 build.xml
146c146
<      <echo message="===== compiling package 'util' ${collections.jar} =====>
<      </echo>
---
>      <echo message="===== compiling package 'utilitaire' ${collections.jar} =====>
>      </echo>
mythos-developer $
```



```
mythos-developer $ cvs annotate build.xml | more
```

```
Annotations for build.xml
```

```
*****
```

```
1.1          (drey      28-Jan-04): <?xml version="1.0" encoding="utf-8"?>
1.1          (drey      28-Jan-04): <!-- =====
=====
1.1          (drey      28-Jan-04):
1.1          (drey      28-Jan-04):      Build file for SmartTools v4 Api - for use with the J
akarta Ant java build tool
1.1          (drey      28-Jan-04):
1.1          (drey      28-Jan-04):      Build Instructions:
1.1          (drey      28-Jan-04):      To build, run build.bat (win32) or build.sh (unix) -
optionally
1.1          (drey      28-Jan-04):      with a target arg as indicated below - in the directo
ry where this
1.1          (drey      28-Jan-04):      file is located.
1.41         (drey      15-Mar-04):      This build.xml uses ./build-properties.xml file.
1.41         (drey      15-Mar-04):      The batch/shell file sets up your classpath and calls
java org.apache.tools.ant.launch.Launcher
1.1          (drey      28-Jan-04):
1.49         (dparigot 09-Apr-04):      Basic target:
1.49         (dparigot 09-Apr-04):      user.create.cmp
1.1          (drey      28-Jan-04):      Decoupage
1.1          (drey      28-Jan-04):      - Defines Global Variables
1.1          (drey      28-Jan-04):      - Configuration
1.1          (drey      28-Jan-04):      - Clean, Init, Present
1.1          (drey      28-Jan-04):      - Target dependances
1.1          (drey      28-Jan-04):      - Code
1.1          (drey      28-Jan-04):      - Web Services
1.1          (drey      28-Jan-04):      - Documentation (javadoc)
1.1          (drey      28-Jan-04):      - Zip et Distribution (demo)
1.1          (drey      28-Jan-04):      - User and Core components
```

# Tagging

- Tag = string marker which is applied to the status of a part of the CVS hierarchy
- Often used to mark a given version of the cvs repository with a string which refers to a version of the software released
- Easily get back specific versions of the software
  - Use/distribute a given released version
  - Reproduce bugs for a given release version
- Only files are tagged
- `cvstag -r 1.2 tagname files`
- `cvstag rtag tagname files` (latest version in the base is tagged)



# Add, delete, and move files

- `cvcs add filenames`
- Produces a message that explains that a commit is necessary
- `cvcs commit -m "addition of files bla..." directories/files`
- Files have to be locally deleted: `rm filenames`
- Produces a message that explains that a commit is necessary
- `cvcs remove filenames`
- `cvcs commit -m "removed files bla..." directories/files`
- Trace of deleted files in Attic directories on the CVS base
- Move = add + delete, with an explicit message to keep a trace of the old names





# Add, delete, and move folders

- As simple as files for addition; do not need a commit command to take effect
- Not possible to delete and/or move folders!
  - Except with “hard” intervention, cf. advanced use of CVS section



# Summary of the main commands

**cv**s [*cv*s-options] *command* [*cmd-options*] [*directories/files*]

<b>cv</b> s <b>import</b>	import files from a local directory to the cvs base
<b>cv</b> s <b>checkout</b>	copy on a local disk a given version of the cvs base
<b>cv</b> s <b>commit</b>	apply the modifications of the local copy to the cvs base
<b>cv</b> s <b>update</b>	upgrade the local copy with a version of the cvs base
<b>cv</b> s <b>add</b>	add a file to the cvs base
<b>cv</b> s <b>remove</b>	remove a file from the cvs base
<b>cv</b> s <b>status</b>	show the status of a local file wrt the cvs base
<b>cv</b> s <b>log</b>	show the different previous commit stages
<b>cv</b> s <b>diff</b>	show lines in local and server files that differ
<b>cv</b> s <b>annotate</b>	show information line by line of a version in the cvs base
<b>cv</b> s <b>tag (or rtag)</b>	put a tag to identify a given version of the cvs base



# Overview

- Collaborative work and version control
- Main CVS user commands
- **Advanced use of CVS**
- CVS server at INRIA Sophia Antipolis



# CVS Administration

- Modules administration
- Usage definition
  - Events control (emails, ...)
  - Watch/edit



# CVS base configuration

- CVSROOT Module (exists by default)
  - cvs checkout CVSROOT
- Configuration files
  - modules, modules definitions
  - cvswrapper binary files control
  - cvsignore list which files CVS has to ignore (\*.o, ...)
  - commitinfo, editinfo, logininfo, notify, rcsinfo, taginfo make it possible to configure actions with respect to specific CVS operations



# Binary files in CVS base

- No real version control (diffs don't work)
- Binaries are corrupted if treated as ASCII files
- Can be done with command line: `cvsv add -kb filename`
- Can be forced for known extensions
  - Configuration file: `cvswrapper`



# cvswrappers - Bloc-notes

Echier Edition Format Affichage 2

```
# This file affects handling of files based on their names.
#
# The -m option specifies whether CVS attempts to merge files.
#
# The -k option specifies keyword expansion (e.g. -kb for binary).
#
# Format of wrapper file ($CVSROOT/CVSROOT/cvswrappers or .cvswrappers)
#
# wildcard      [option value][option value]...
#
# where option is one of
# -f            from cvs filter          value: path to filter
# -t            to cvs filter            value: path to filter
# -m            update methodology      value: MERGE or COPY
# -k            expansion mode           value: b, o, kkv, &c
#
# and value is a single-quote delimited value.
# For example:
# *.gif -k 'b'
# *.jar -k 'b'
# *.zip -k 'b'
# *.pdf -k 'b'
# *.ppt -k 'b'
# *.doc -k 'b'
# *.gz -k 'b'
# *.ps -k 'b'
# *.jpg -k 'b'
# *.gif -k 'b'
# *.png -k 'b'
# *.tif -k 'b'
```

# Recursion with CVS

- Check each command to see if it is recursive or not
  - Add is not recursive, update is recursive, ...
- Read the cvs manual (note we are in the « advanced section » of the expose)





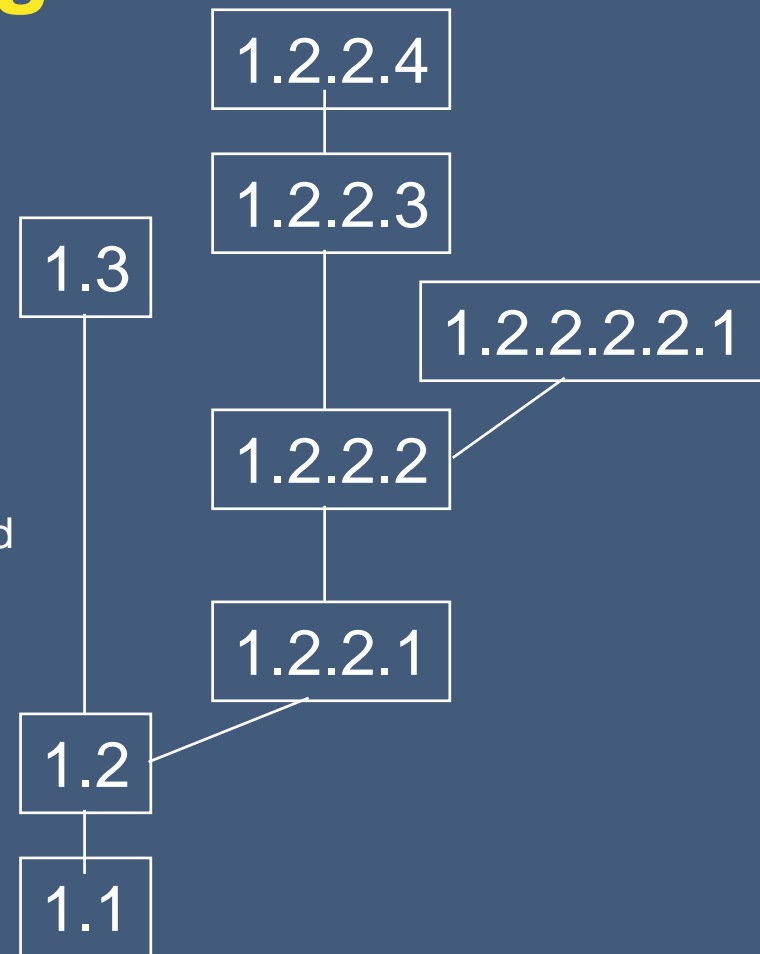
# Windows users

- Several CVS clients: webCVS, winCVS, ...
- Possible use of CVS with ssh under Windows via putty (Windows ssh client) → contact me
- Problems with line ending ... always edit files on the platform where the checkout has been done!



# Branching

- Branching is easy... merging is not!
- When ?
  - Before a release (“freeze” the code in a branch)
  - For parallel development that will not be integrated directly
    - For someone developing during a training period
    - For some “research code”
    - For a specific application (e.g. dedicated to a given client)





# Direct manipulations on the server



- To restructure a repository (e.g. move or delete folders)
- To check for possible problems
- For instance, administrators of a CVS repository at Sophia can connect to the server via ssh:
  - `ssh cvs-sop.inria.fr`
  - `cd /CVS/SmartTools`



# Overview

- Collaborative work and version control
- Main CVS user commands
- Advanced use of CVS
- CVS server at INRIA Sophia Antipolis



# Administred by the semir

- Semir page about the INRIA cvs server:

<http://www-sop.inria.fr/semir/serveurs/cvs/>

- Hotline SEMIR to open a new repository:

- <http://www-sop.inria.fr/semir/support/>
- “Logiciels/Outils” then “Agenda, CVS, FTP, MYSQL”
- Chose “CVS” and “Creation”
- ...



*Sophia  
Antipolis*

# ssh access

- Good level of security for the authentication
- <http://www-sop.inria.fr/semir/serveurs/cvs>
- In actual using, only few differences:
  - `cvs -d :ext:drey@cvs-sop.inria.fr:/CVS/dream checkout web`



# Administrative tasks

- Standard CVS administration (cf. “CVSROOT” Module)
- “Admin” module
  - automatically in the repository when created by the SEMIR
  - 3 files: group-def, modules-def, passwd-def
- **group-def**: define and describe groups of users
- **modules-def**: define access rights for each module wrt groups (ssh, anonymous)
- **passwd-def**: define and describe accounts (username, crypted password and email ) for non INRIA users



# References and links

- **CVS:** <https://www.cvshome.org/>
  - <https://www.cvshome.org/docs/manual/>
  - <https://ccvs.cvshome.org/fom/fom.cgi>
- **GUI:**
  - <http://www.twobarleycorns.net/tkcv.html>
  - <http://www.wincvs.org>
  - <http://www.tortoisecvs.org>
  - <http://www.jalindi.com/igloo>
  - <http://www.jcvs.org>
- **INRIA server:** <http://www-sop.inria.fr/semir/serveurs/cvs/>
  - <http://www-sop.inria.fr/semir/serveurs/cvs/#p3>
- *CVS Configuration et mise en œuvre.* Frédéric Lepied. O'Reilly Eds.  
(available at the documentation centre of INRIA Sophia: ref. O356 )





# Closing remarks & Questions

- Use version control systems **from the start**
  - When working alone or in a group
  - For development projects or scientific papers
  - When tag released versions
- Handy FAQ's on the CVS home site

**Next seminar is next year! January 13th on Refactoring**



*Sophia  
Antipolis*