

# Instructions for Source Code Repository (Version Control)

document created on May 2, 2013, updated on June 3, 2013

If you have any problems with this document, contact Ahmet Erdemir, [erdemira@ccf.org](mailto:erdemira@ccf.org).

The project source code repository is a centralized repository based on [Subversion](#). It is intended to facilitate collaborative development while keeping an eye on version control. In this document, the team members can find brief instructions to facilitate their use of the repository within the context of project requirements. Members of the project have full permissions to access the repository and change it. Appropriate links are provided for more detailed help on [Subversion](#).

## *It is IMPORTANT for team members to use the source code repository because*

- it allows concurrent and collaborative development of documents;
- all documents are under version control:
  - mechanisms are implemented to resolve conflicting changes,
  - it is always possible to revert (undo) files to previous versions;
- documentation activities of individual team members and the whole project are clear and accessible by all team members and potentially by everyone, resulting in:
  - accurate acknowledgment of activities,
  - facilitated dissemination;
- mature versions of the documents can be provided as release versions.

## Table of Contents

|   |   |
|---|---|
| Repository Location.....                    | 2 |
| Browsing.....                               | 2 |
| Full Access with Subversion Client.....     | 2 |
| Repository Organization.....                | 2 |
| Subversion Terminology.....                 | 3 |
| Interacting with the Repository.....        | 3 |
| Checking-Out a Local Copy.....              | 3 |
| Updating a Local Copy.....                  | 5 |
| Committing Local Changes to Repository..... | 5 |
| Creating Files and Directories.....         | 6 |
| Advanced Actions.....                       | 7 |
| Comparing Versions.....                     | 7 |
| Undoing Changes.....                        | 7 |
| Moving Files or Folders.....                | 7 |
| Helpful Links.....                          | 7 |

## Repository Location

### **Browsing**

Project repository can be "viewed" online using any web browser at

<https://simtk.org/websvn/wsvn/cpms/>

The same area can be browsed by accessing the *Source Code Repository* section under the *Advanced* link at the project site. General public has read access to all folders and files in the repository, whereas being a team member is necessary to have full access privileges (read+write). While online browsing is useful to understand repository structure and to quickly view some files and revision history, team members who want to work on the files/folders should utilize a client software for full access (see [Helpful Links](#)).

### **Full Access with Subversion Client**

Repository can be accessed through a [Subversion](#) client (see [Helpful Links](#)), which will allow team members

- to make local copies of the whole repository (or its subfolders)
- to update their local copies with most recent versions
- to provide their changes back to the repository

For these purposes, the team members should direct their [Subversion](#) client to

<https://simtk.org/svn/cpms>

A variety of clients can be used to interact with the repository. Some useful clients with graphical interfaces include but not limited to

- RapidSVN (<http://www.rapidsvn.org/>)
- TortoiseSVN (<http://tortoisesvn.net/>)

## Repository Organization

The repository contains a **dynamic** and **collaborative** development environment. While there are not any rigid rules on the organization of content, a few guidelines may help team members to create additional folders and files to prepare documents as part of project activities. The main outline of the repository structure (as of May 2, 2013) is illustrated below.

```
.
├── doc
│   ├── abstracts -- abstract submissions to meetings
│   ├── bios -- biosketches for Committee Members and Advisory Council
│   ├── help -- instructions to use various collaboration tools
│   ├── img -- useful images for documentation and visibility
│   ├── invitation_letters -- letters to recruit Committee Members and Advisory Council
│   ├── minutes -- summary of Committee Member and Advisory Council meetings
│   ├── notes -- various raw information
│   ├── posters -- posters to promote visibility
│   └── presentations -- presentations to promote visibility
```

Subfolders for individual tasks can be created under a suitable main folder. Please use **specific, descriptive, and short** names for subfolders.

Details of creating new folders in the repository and working with repository content are described below. There is no right or wrong way to organize folder contents. As long as the folder is under version control, the overall structure and files can be changed without loss of information and continuity.

It is advised to subscribe to the mailing list [CPMS-commits](#) that can be found in the project site under *Advanced->Mailing Lists*. This will ensure that team members get e-mail notifications when others change the repository contents.

## **Subversion Terminology**

### **checkout**

making a local copy of the online repository by downloading the repository contents

### **update**

updating the local copy of the repository; changes conducted on the local copies of files/folders are not overwritten

### **add**

adding a file or folder under the local copy structure to the online repository definition;

### **add recursively**

adding a folder under the local repository copy structure to the online repository definition; including all folder contents

### **commit**

uploading files/folders (added or changed, if already existing) in the local copy to the online repository

### **revert**

undoing changes conducted on the local copies of files/folders by downloading desired versions from the repository; overwrites the local files/folders

### **diff**

compare two different versions of a text file; requires utility software for file comparison

## **Interacting with the Repository**

Examples to interact with the repository are provided using [RapidSVN](#), a [Subversion](#) client. You may want to download this client or any other to test. Other clients work similarly using the same terminology.

### **Checking-Out a Local Copy**

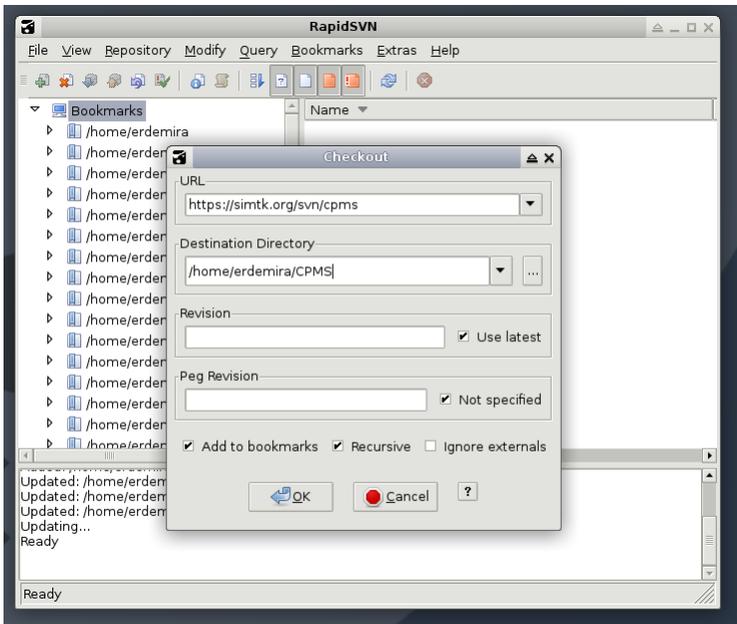
Using [RapidSVN](#), a copy of an online repository (or parts of it) can be downloaded to the local computer from the menu:

- Repository -> Checkout

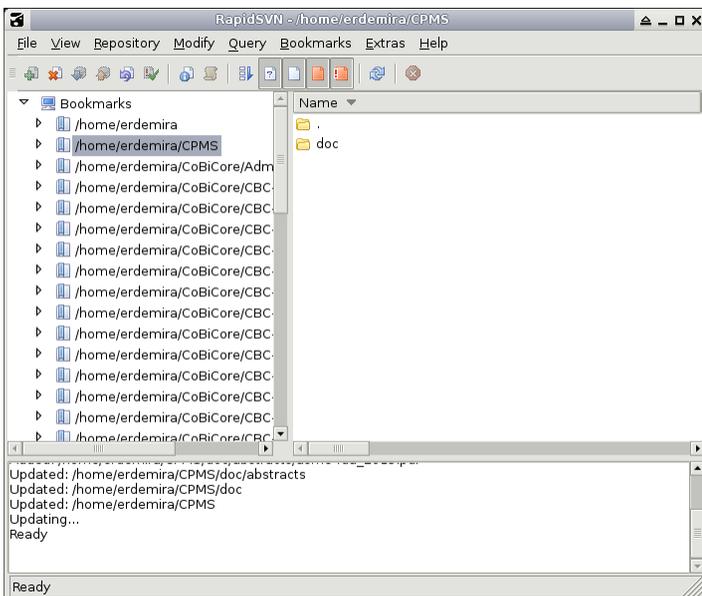
Checkout process requires:

- the remote address (URL) of the repository
- destination directory on the local computer
- revision (by default it is set to latest)

By default, the local destination is added to the bookmarks for quick access and the repository contents are downloaded recursively. The example below illustrates checking out the main folder of the project and all its subcontents to a local directory on the computer in `/home/erdemira/CPMS`:



Once checked-out, one can highlight the added bookmark to the local copy to see its contents:



**For full access (read+write) to the repository, team members need to login using their SimTk account.** When authentication is requested, enter your SimTk username (use only lowercase letters) and password. Since the repository has read access for everyone, login information will not be asked during check out. It will be asked during committing local changes to repository (see below). It is also possible to provide login information

for an already checked-out repository, by selecting the bookmark of the local copy and the *Login* command in the right-click menu.

## Updating a Local Copy

Updating a local file or folder with a desired version in the repository can be done by selecting the file/folder and by following either of these steps:

- Modify -> Update in the main menu
- Update command in the window following right mouse click

By default, the latest version will be used to update local files and folders recursively, including subfolders and files.

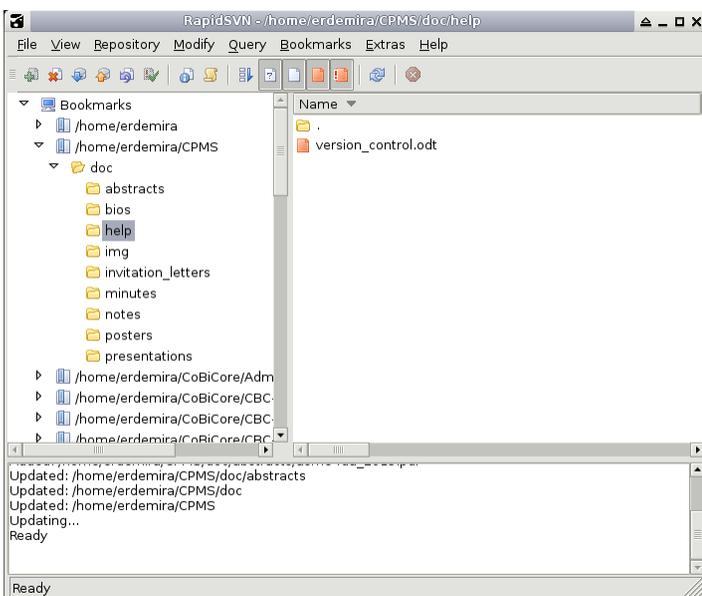
**Always update the local copy with the most recent version of the repository before changing files. Another team member may have worked on the same files.**

## Committing Local Changes to Repository

After changing the files locally, the online repository can be upgraded by selecting these files (the local copies) or the folders, and by following either of these steps:

- Modify -> Commit in the main menu
- Commit command in the window following right mouse click

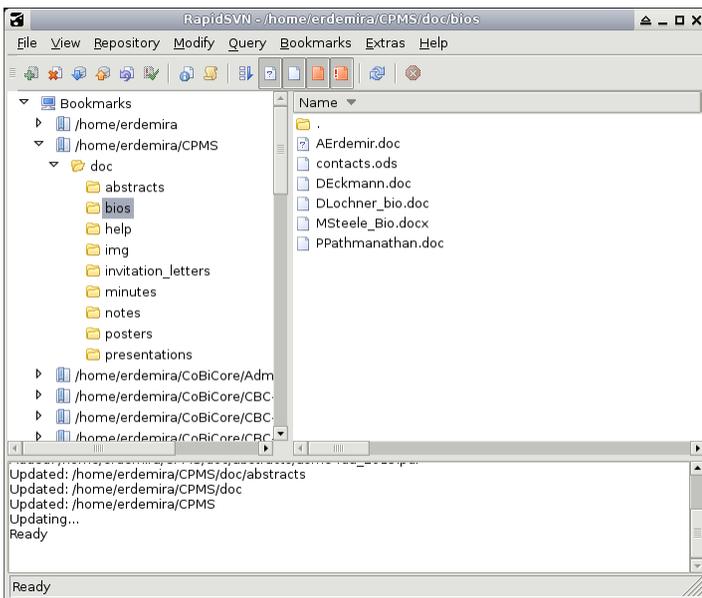
In the example below the local copy of the *version\_control.odt* file was changed. The red shading of the icon indicates that the file is different than the last version updated from the repository. When this file is committed, the shading of the icon will turn white. During *Commit* process, providing a log message will be helpful in prospective reviews of the time history of the repository and individual files.



## Creating Files and Directories

Files and directories can be created under the local copy of the repository as during daily operations. However, to ensure these files/folders are part of the repository, the remote (online) repository need to be informed about their existence. This can be accomplished by the *Add* or *Add recursive* operation. Note that this procedure only indicates that the new files and/or folders will be part of the repository. To upload these, a *Commit* operation should be followed, as described above.

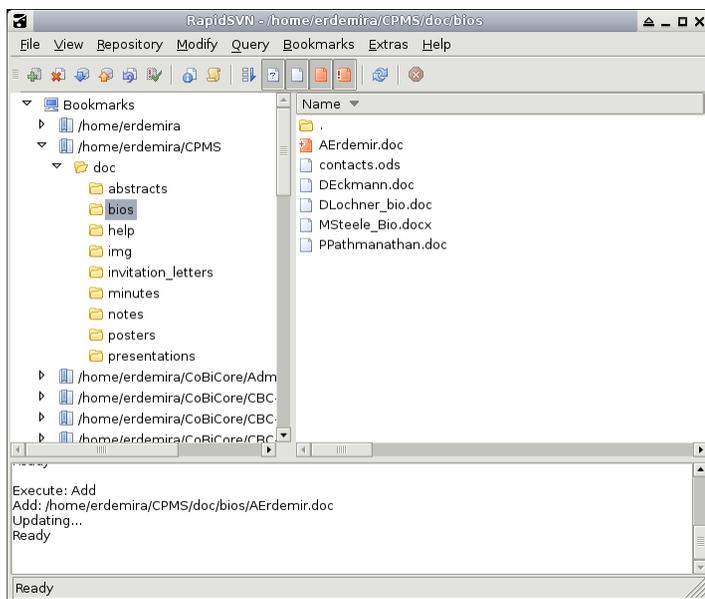
In this example, a file named *AErdemir.doc* was created under the *bios* folder on the local computer. When the folder is browsed using [RapidSVN](#), a question mark can be seen in its icon. This indicates that this file does not belong to the repository yet.



Selecting this file and employing

- Modify -> Add in the main menu, or
- Add command in the window following right mouse click

indicates that the file will be part of the repository. Consequently, the shading of the icon will turn red, with a '+' sign. A *Commit* operation uploads the file to the repository.



## Advanced Actions

### Comparing Versions

Text files in the repository can be compared to previous versions using a *Diff* tool. In the main menu at View -> Preferences -> Programs tab, a Diff tool needs to be defined, e.g., [Meld](#), [WinMerge](#).

- Query -> Diff in the main menu
- Diff command in the window following right mouse click

### Undoing Changes

If the changes in the local copy of the repository are undesired, and an undo operation is necessary, the *Revert* functionality can be used:

- Modify -> Revert in the main menu
- Revert command in the window following right mouse click

By default, the latest version will be used to overwrite the local changes.

### Moving Files or Folders

- Modify -> Move in the main menu
- Move command in the window following right mouse click

## Helpful Links

- Documentation for Subversion (<http://svnbook.red-bean.com/>).
- **Software Carpentry** (<http://software-carpentry.org/>): Useful information on software development for scientists, particularly on **version control**.
- **RapidSVN** (<http://www.rapidsvn.org/>): RapidSVN is a cross-platform, free and open source client for **Subversion**.

- **TortoiseSVN** (<http://tortoisesvn.net/>): TortoiseSVN is an operating system specific, free and open source client for **Subversion**.
- Also check [http://en.wikipedia.org/wiki/Comparison\\_of\\_Subversion\\_clients](http://en.wikipedia.org/wiki/Comparison_of_Subversion_clients).