Software Build Automation Tools

Laboratory Exercises 3

Subject
Version Control Management in Subversion (SVN)

Author:
PhD Hubert Zarzycki

The scope of laboratories

Introduction

This laboratory session introduced SVN as a version control tool and guides through basic and advanced operations. Understanding SVN workflows, collaboration, and repository management is essential for software development in team environments. Mastery of these concepts will enhance ability to work efficiently in professional projects.

By completing these exercises, you will be able to:

- Set up and use an SVN repository.
- Perform essential version control operations.
- Collaborate on software projects using SVN.
- Manage branches and releases effectively.

Further exploration of advanced SVN features, such as hooks, automation, and integration with CI/CD pipelines, is encouraged to deepen understanding of version control practices.

SVN

Version control systems (VCS) play a crucial role in modern software development, ensuring that changes to the codebase are tracked and managed efficiently. One of the popular centralized version control systems is *Apache Subversion (SVN*). This laboratory exercise introduces to SVN using the *Assembla.com* platform, covering repository setup, basic operations, and collaborative workflows.

The primary objective is to provide users with practical experience in managing software versions using SVN. By the end of this session you will have a foundational understanding of version control, enabling them to use SVN for software development projects efficiently.

Getting Started with Assembla

To work with SVN, a remote repository is required. *Assembla* offers free and paid repository hosting services, making it an excellent choice for practicing version control management.

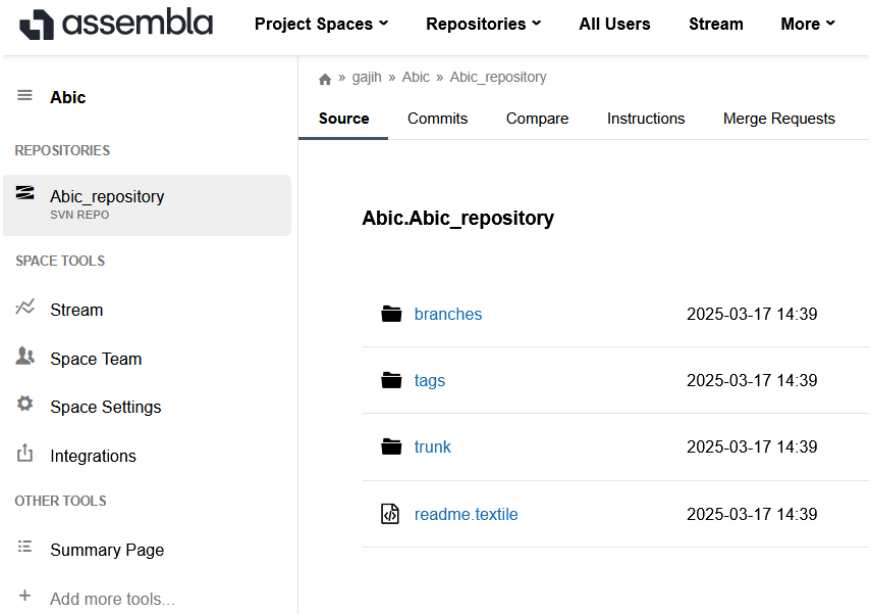
Software Build Automation Tools

1. Creating an Assembla Account

- Open a web browser and navigate to https://www.assembla.com.
- Click Sign Up and create an account using an email address.
- Verify the email address and log in to the Assembla platform.

2. Setting Up a Repository

- After logging in, navigate to "Workspaces" and select "Create a New Space".
- Enter a name for the project and choose "SVN Repository" as the version control type.
- Click "Create Space" to initialize the repository.
- Once created, navigate to "SVN Repository" to find the repository URL. Below is a screenshot of an example SVN repository on Assembla.



Installing a Subversion Client

To interact with the repository, an SVN client is required. Recommended options include:

- TortoiseSVN (Windows)
- RabbitVCS (Linux)
- SVN command-line tools (available for all platforms)

Installation steps:

- Download the preferred SVN client from its official website.
- Follow the installation instructions.
- Verify the installation by running svn --version in the command line.

If administrator privileges are not available on the computers, you can still complete the tasks using **web-based SVN clients** or **portable SVN solutions**. Here are some possible approaches:

- Using *Portable SVN Clients*

    - Portable Smart SVN (https://www.smartsvn.com/): SmartSVN offers both an installation and a portable version, but the portable version may require purchasing a license after the free-trial period. Here is a link to the portable version 14.5.0 SmartSVN portable.
    - *Portable TortoiseSVN:* A version of TortoiseSVN that runs from a USB drive without installation.
    - *SVN command-line portable versions*: Some distributions allow running without installation (e.g., standalone binaries).
- Using *Assembla's Web Interface*
    - Assembla provides a built-in web-based file editor that allows users to add, modify, and commit files directly through the browser.
    - you can create and edit files, commit changes, view repository history, and manage branches/tags without installing an SVN client.
- Using an *Online SVN Client*

  Some services offer browser-based SVN clients that allow interaction with repositories without requiring software installation. Examples include:
    - SVNWebClient (https://svnwebclient.tigris.org/)
    - RISE SVN Web Client (if available for institutional use)
- Using a Cloud-based Virtual Machine
- If the institution provides access to cloud-based development environments (e.g., Google Cloud Shell, Gitpod, AWS Cloud9), users can install and use SVN remotely.

Basic SVN Operations

After setting up the repository, basic operations can be performed.

1. Checking Out a Repository

To retrieve a working copy of the repository:

```
svn checkout <repository_url> <local_directory>
```

Example:

```
svn checkout https://subversion.assembla.com/svn/myproject/ myproject
```

2. Adding and Committing Files

- Navigate to the working directory:

```
cd myproject
```

- Create a new file and add content:

```
echo "Hello, SVN!" > readme.txt
```

- Add the file to version control:

```
svn add readme.txt
```

- Commit the file to the repository:

```
svn commit -m "Added readme file"
```

3. Updating the Working Copy

To fetch the latest changes from the repository:

```
svn update
```

4. Viewing Repository History.

To check the commit history:

```
svn log
```

5. Reverting Changes

To undo local changes before committing:

```
svn revert <filename>
```

Example:

```
svn revert readme.txt
```

Tasks

1. For the tasks create a report including:
   - Authors' names
   - Repository link
   - Steps performed during the lab.
   - Screenshots of key operations (checkout, commit, update, merge, etc.).
   - Description of any challenges faced and solutions applied.
   - Final state of the repository.

The report should be submitted via the e-learning platform as a DOC or PDF document or archived project files.

Initializing a Repository and Committing Changes

2. Create an Assembla account and set up an SVN repository.
3. If possible install an SVN client on the local machine.
4. Check out the repository to a local directory.
5. Add a new file (project.txt) with a brief project description.
6. Commit the changes with an appropriate commit message.
7. Verify the commit using svn log.

Collaborating with Version Control

8. Form teams of two or more students.
9. Each team member should check out the same repository.

10. Modify the project.txt file by adding their name and role.
11. Commit the changes and observe possible conflicts.
12. Resolve conflicts manually and commit the resolved file.
13. Verify the final version in the repository.

Working with Branches

14. Create a new branch for feature development:

```
svn copy <repository_url>/trunk <repository_url>/branches/feature-x -m "Crea-
ting feature branch"
```

15. Check out the branch locally:

```
svn checkout <repository_url>/branches/feature-x
```

16. Modify files in the branch and commit changes.
17. Merge the branch into the trunk after completing development:

```
svn merge <repository_url>/branches/feature-x
```

18. Commit the merged changes.
19. Delete the branch if no longer needed:

```
svn delete <repository_url>/branches/feature-x -m "Deleting branch after merge"
```

Tagging Releases

20. Create a release tag from the trunk:

```
svn copy <repository_url>/trunk <repository_url>/tags/release-1.0 -m "Tagging
release 1.0"
```

21. Verify the tag exists in the repository.
22. Checkout the tag and ensure it is a snapshot of the trunk.

23. Please submit to the e-learning platform a report summarizing the work, including challenges faced and solutions applied. Include a link to the repository.

Literature

Loeliger, J., & McCullough, M. (2012). *Version Control with Git: Powerful tools and techniques for collaborative software development.* O'Reilly Media.
Chacon, S., & Straub, B. (2014). *Pro Git.* Apress.
Rubalcaba, C. (2021). Git for Programmers*: Master Git for Effective Development and Deployment.* Apress.
Collins-Sussman, B., Fitzpatrick, B. W., & Pilato, C. M. (2008). *Version Control with Subversion*. O'Reilly Media.
Mahler, A. (2015). *Practical Subversion*. Apress.
Pérez, R. M. (2013). *Apache Subversion 1.7 Quick Reference Card*. Packt Publishing.
GIT Documentation - *https://git-scm.com/doc*
GitHub Learning Lab - *https://lab.github.com/*
Atlassian GIT Tutorials - *https://www.atlassian.com/git/tutorials*
GitHub Actions Documentation - *https://docs.github.com/en/actions*
Jenkins CI/CD Documentation - *https://www.jenkins.io/doc/*
Apache Subversion Official Site*: https://subversion.apache.org/*
Subversion Book (Free Online Guide): *https://svnbook.red-bean.com/*
TortoiseSVN (Popular SVN Client for Windows): *https://tortoisesvn.net/*
Assembla SVN Repository Service: *https://www.assembla.com*