



Let's Help Bo Project SVN Policy

Version 1.1

LHB Project	Version: 1.1
SVN Policy	Date: 2013-1-10

Revision History

Date	Version	Description	Author
2012-11-09	1.0	Initial Draft	Petar Stojanac
2013-1-10	1.1	Final Version	Petar Stojanac

LHB Project	Version: 1.1
SVN Policy	Date: 2013-1-10

1. Introduction

1.1 Purpose of this document

The purpose of this document is to establish a set of rules for using the SVN repository. It's also to provide a set of helpful guidelines and a description of the practices and policies in order to minimize the amount of conflicts that will occur. The folder hierarchy will also be described.

1.2 Document organization

The document is organized as follows:

- Section 1, *Introduction*, describes contents of this guide
- Section 2, *Guidelines*, describes the guidelines
- Section 3, *Folder Hierarchy*, describes that folder will be created in the repository and how they will be organized

1.3 Intended Audience

The intended audiences are the team members of the LHB Project

1.4 Scope

The document addresses all the issues regarding the handling of the SVN such as: the best practices when using the SVN, how to use the SVN (adding, deleting, modifying), information about doing the backup of the repository and the directory hierarchy.

1.5 Definitions and acronyms

1.5.1 Definitions

Keyword	Definitions
update	SVN command to check if there were changes made by other user on the files in the repository
commit	SVN command to upload the files that the user changed to the repository
merge	SVN command to merge different version of one file in a new file
add	SVN command to add files to the queue of files to be added to the repository

1.5.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
NTR	Nothing to Report. There is no information to a specific topic available or necessary.
LHB	Let's Help Bo
SVN	Subversion

1.6 References

Ignore files and folders - <http://stackoverflow.com/questions/588077/what-visual-studio-files-should-be-ignored-by-subversion-to-minimize-conflicts/>

LHB Project	Version: 1.1
SVN Policy	Date: 2013-1-10

2. Guidelines

Here are some of the guidelines and best practices when using the SVN repository. If these are followed throughout the implementation phase the number of conflicts should be minimized.

2.1 Workflow

Describes which actions should be performed and in what order when committing. Assumption is that the repository was already checked-out into a local folder.

- Before any work is done the user needs to *update* the repository in order to get the newest files
- After the update the user is free to edit the files
- After the user is done editing he needs to *update* again in case there were additional changes made by other team members in the time period he was editing
- If there is a conflict the user can merge the conflicted version to a new version (he can check for the differences between the two files using *Diff* options) which will contain both his changes and the changes of the other team member or if he is not sure of the changes he should NOT commit and he should contact the SVN manager instead. Using the differences view user can also choose whose changes he will take (choose "Other" if not sure).
- If there is no conflict or if the user chose to merge the files he will proceed with committing the files. When committing the user must follow the guidelines for a proper way to commit (Section 2.2)
- **When committing** make sure all the necessary files are committed. If the user created some **new** folders and files that were not on the SVN before he needs to *add* them first and then commit otherwise they will not be uploaded to SVN, other users will not see them and they will get missing references errors. Also see the section about files that can be ignored (Section 2.3)

2.2 Committing

- **Make sure** that your code compiles and does not have any errors before you commit it to the SVN
- **Do not commit blindly**, test the changes you made by running the application before you commit anything
- **Double check for conflicts** before committing the changes by updating before committing. The differences between the files that are on the SVN and your changes to them can be seen using the *Diff* option.
- **Don't leave the message log empty**. User must always leave a short description of the changes he made in his commit. He will describe (short) on what he was working on and if any bugs were fixed. This is important so that it can be tracked later on if some errors occur and where those errors were made. And also to be able to revert to a state before the error occurred.
- **Think before committing and merging ("Commitiquette")**. If the users commit/merge is going to change someone else's code contact them first and let them know. Do not change the code of another coder if he doesn't agree to it first.
- **After the committing** take appropriate measures to see if the application works as it should. Try to run it and ensure there are no errors and that everything is committed. In case there are errors and the user committed wrong files he must contact the SVN manager and together with him fix the damage. Even if the user is able to fix the error he must inform the team and the SVN manager.

LHB Project	Version: 1.1
SVN Policy	Date: 2013-1-10

2.3 Files and folders that can be ignored

When using subversion and visual studio there are certain files that can cause conflicts and they should be put on the SVN ignore list. Those are usually some output files are user specific visual studio files. Here is a list of such folders and files:

- bin directory
- obj directory
- *.suo files - user specific
- *.user files - user specific
- Thumbs.db - MS thumbnail file

3. Folder Hierarchy

- Code
 - LHB
- Documentation
 - Deliverables
 - Policies

Deliverables folder will hold our project deliverables once their initial version is finished. Documentation folder might be slightly altered to accommodate for other deliverables such as Minutes of meeting documents and week reports if we choose to put them on SVN.

4. SVN Manager

Team member Petar Stojanac is in charge for the SVN. His duties are helping to resolve the conflict and doing regular backups of the repository.

5. Backups

Backups will be made by the SVN Managers every 3 days in form of archived file and they will be uploaded to Dropbox.