SVN Mini How-To:

"SVNMerge says there is a conflict, and now Exper doesn't work. What do I do?"

1. Don't panic.
2. Right-click on the C:\lab folder and select "SVN Resolved…"
3. This opens a window listing all the files with conflicts. At this point you might be able to guess who else has been editing those matlab files, and you might want to communicate with them about what has changed. Or you can just proceed, since you'll always be able to revert later.
4. Right-click on a file and go to "Edit Conflicts"; this will open a diff viewer window.
5. The diff viewer window has three panels:
    a. The left panel ("theirs") shows the working base, the version of the file on the repository.
    b. The right panel ("mine") shows the working copy (your version) of the file.
    c. The bottom panel ("merged) shows the 'merged' copy, the version you will be creating to resolved the conflicts.
    d. The differences between the two file versions are marked in red.
    e. note: only numbered lines are actually in the file
6. For each red text block, decide which version to use (theirs, mine, or both). Right click on the text block and select which version to use. If there are multiple red text blocks, you could also choose to use one entire file (theirs or mine). The result is shown in the bottom panel (merged version). You can also just edit the bottom panel directly if you want to.
7. When the bottom panel (merged version) looks right to you, click on the save button. Then close the window
8. Back in the Resolve window, click OK. Another Resolved window should pop up saying "finished." Click OK again.
9. Run SVNMerge again. It's on the desktop and in C:\lab.

Daily use: using SVNMerge.bat

The process of updating the working copy and committing changes (see below for details) has been made smoother by the creation of a little batch file called SVNmerge.bat. This batch file will take care of updating before and after a commit and creates a log of the output from its last run in SVNmerge.log. Just double click the file to keep your computer up to date.

Installling a new SVN client and checking out exper in Wehr Lab:
(For example, when setting up a new rig computer)

1. Download and install the latest version of Suversion command-line client at:
http://www.collab.net/downloads/subversion/
you will need to log into collabnet, you can use user wehrlab, password wehrA1

2. Get and install the latest version of TortoiseSVN (the Windows GUI for SVN) at:
http://tortoisesvn.net/downloads

3. Checkout
1. Make sure the C:\lab folder no longer contains the exper2.2, labtools, and E2Scripts folders. If it does, delete them.

2. Right click on the C:\lab folder and go to “SVN Checkout…”
3. In the URL field type “svn://184.171.85.38/exper/trunk”
This will download a copy of the files in the exper repository to the C:\lab folder (can take a while).
This folder is now what referred to as a “working copy” in svn terms.

Sometimes checkout will fail (time out) even when the URL is correct. Not sure why this happens, but the following have sometimes worked in this case.
http://184.171.85.38/exper/trunk
svn://blister.uoregon.edu/exper/trunk

Relocating:
If the ip address of the server changes (for example, when we moved it to the new building) you can get your working copy to point to the new server using the relocate command. Example: the new SVN server is blister.uoregon.edu. Open a shell and navigate to the working copy:
$cd labtools
$svn relocate svn://blister.uoregon.edu/exper/trunk/labtools
that's it. repeat with exper2.2 and E2Scripts.

Installing the SVN **server** in Wehr Lab:
(Note: The server is already installed, so you probably don't need to do this)

Installing Subversion and Starting a Server

First, download latest version of SVN (1.51 as of writing) at:
http://www.collab.net/downloads/subversion/

Get the latest version of the Windows GUI for SVN (TortoiseSVN) at:
http://tortoisesvn.net/downloads

Install the programs. Try using a command window (cmd) by typing “svn”. If the program isn’t found make sure the /bin subfolder of the installation directory of Subversion is in the path.

Open a command prompt and type:

svnadmin create "C:\experSVN"

Navigate to the folder we just created. Within that folder, uncomment the following lines in the /conf/svnserve.conf file:

[general]
anon-access = read
auth-access = write
password-db = passwd

Next, uncomment these lines in the /conf/passwd file:

[users]
harry = harryssecret
sally = sallyssecret

You should change the “users” section to contain the needed users and passwords (user = password).

You will then need to create a windows service to run the SVN server.

Use the command:

sc create svnserve binpath= "c:\svn\bin\svnserve.exe --service –-root C:\experSVN displayname= "Subversion Server" depend= Tcpip

Then use:
sc config svnservice start= auto
net start svnservice

and you should have an SVN server running using C:\experSVN as its repository.

Manually Committing Changes and Checking Differences

The working copy (WC from now on) directory will show a green check mark (as will all of its contents) now that it is being tracked by SVN. The files SVN is not tracking (unversioned files) in the C:\lab directory will all have question marks.

After editing a file it the icon will change to show a red circle with a white exclamation point, don’t be alarmed this is normal.

When your'e done making changes and you want to commit them to the repository, right-click on the C:\lab folder and go to “SVN Commit…”.

This will open a window that displays all of the files you have modified since the last commit. If you are not sure about the changes made to some files then you can uncheck them.

When you are done click OK.

Notes:
If you want to check the line-by-line differences between the current file in the repository (known as a working base copy) you can right-click on the file and go to “View in diff viewer”. This will show a side-by-side of differences between the two versions.

Warning:
When committing make sure that you do not check files or folders that like “Data” that are unversioned and need to stay that way. To help yourself, you can uncheck the “Show Unversioned Files” option at the bottom of the window.

Adding Files

If you have created a file that you want to add to the repository for others to use and edit you can right-click on the file (or one of a group of selected files) and go to “SVN Add…”. This is most common with protocols that get created with one of the “Make” commands and that you want to use on another rig.

These files will not be added to the repository until the next commit.

Updating

Updating will download any new changes to the repository onto your WC. You should update before you make any changes, before you commit, and even after you make changes.

To update, right click on C:\lab and go to SVN Update…, this will pop up a window showing the files being changed (or not) and any conflicts – see section VII.

Identifying and Resolving Conflicts

A conflict occurs when changes made by you and changes committed since your last update overlap.

If you have conflicts, the WC directory and any directory containing the conflicting files will be marked with a yellow triangle with a black exclamation point.

To begin resolving conflicts, right-click on the C:\lab folder and go to “SVN Resolved…”

This opens a window listing all the files with conflicts. Right-click on a file and go to “Edit Conflicts”; this will open a diff viewer window -- for info on editing files using the diff viewer see section VII.

Once all conflicts have been resolved click OK.

Now recommit (or run SVNmerge.bat again).

Notes:
If you used the batch file, it does not correct conflicts it postpones them. You will still have to resolve the conflicts. However, you can look in the SVNmerge log for lines beginning with “C” to find the files with conflicts.

Using the Difference Viewer

A difference viewer (diff viewer from now on) is a special text editor to view different versions of a file simultaneously and is used to resolve conflicts (merge versions of files to together).

You can get a two panel diff viewer by right clicking on a file and going to View Diff.

When you click “Edit Conflicts” like in section VII you will bring up a diff viewer window with three panels.

The left panel (by default) shows the working base, the version of the file on the repository.
The right panel shows the working copy (your version) of the file.
The bottom panel shows the ‘merged’ copy, the copy you will be creating to resolved the conflicts.

The diff viewer ‘knows’ where the conflicting sections of text are and highlights them. The sections are ‘smart’ in that they are separated out by the diff viewer and are treated as whole blocks. When you right click on one of these sections it will select the entire section and give you a menu to use (in the merged copy) the version of the file in the panel you right-clicked in, or the other panel, or to use both (with the choice of which will come first).

Notes:
If you have a large section in conflict and want to combine parts from both versions its best to start by choosing to use both versions. Then you can get all the changes in one place and edit just the merged copy the rest of the time. You can even save and exit at this point and open the merged copy, which is now your working copy, and edit it in, say, the MATLAB editor.

Some lines may be crossed out in the viewer. These are lines that were removed in a previous version of the file. These sections are treated like any other section.

Viewing Changes Over Time

Looking at changes a repository has gone through can help in programming/bug fixing.

To view the changes to the repository as a whole you can use right-click > ‘SVN Revision Graph’. This brings up a graphical representation of the history of the repository with revisions represented as boxes or circles. This isn’t very useful for a repository without any branches.

However, you can right click on one of the revisions and go to “Show Log” to see a list of logged changes to the repository up to that revision.

In this window you can left click on one of the logs (in the top panel) to see the log message for that revision as well as any files that were changed/added/deleted in that revision in the bottom panel.

By right clicking and going to “Show Changes” you will open a diff viewer window with a standard two panels like you have directly right clicked on file in explorer.

You can see a color coded history of the file by going to “Show changes as a unified diff”.

This diff viewer window has only a single panel but shows a color coded file with all revision changes to it.