# Merging from stable to trunk

1. Check-out trunk
2. Move to the root directory where you have your clean copy of trunk
3. Using the command line at the trunk directory, run php doc/devtools/svnbranchupdate.php branches/**version_number**
    - Example: php doc/devtools/svnbranchupdate.php branches/12.x
    - Make sure your version of trunk is up to date and there are no uncommitted changes
        - You should not even update composer, only svn up - it is best if this is a totally clean, dead copy of trunk.
    - This command is pointing to the branch on the repository, not your copy
4. Fix conflicts and commit using svn commit -F svn-commit.tmp.
5. If there are conflicts that you cannot fix, try svn revert ./ to reverse the merge and notify the dev mailing list
    - Whatever you do, don't just commit parts of the changes as that results in changes getting missed between the versions

**example**

```
[demotw@alpha trunk]$ php doc/devtools/svnbranchupdate.php branches/12.x
Verifying...
Updating...
Merging...
svn: Write error: Broken pipe
--- Merging r35997 through r36007 into '.':
U    lib/wiki-plugins/wikiplugin_addtogooglecal.php
After verifications, commit using `svn ci -F svn-commit.tmp`
```

You can ignore the message svn: Write error: Broken pipe.

If you are new to this and don't want to have a conflict to resolve: wait for someone else to merge and then do your changes on 12.x and right after, merge to trunk.

If you are outside this period, you can manually merge a commit: http://dev.tiki.org/Merge a commit to trunk" class="wiki wikinew text-danger tips">Merge a commit to trunk

# Q & A

# Q: How to test if it is a dead trunk copy ?

A: ...

# Q: How do I check there is a conflict ?

A: ...

# Q: How to resolve conflicts ?

A: In GNU/Linux, you may be able to use "Meld" as an external tool to resolve conflicts, and configure subversion to launch it when needed with the option **l**. See:

- http://codingundertheinfluence.blogspot.com.es/2010/01/using-meld-as-your-external-diff3.html[1]

line edited in the /root/.subversion/config (and /home/xavi/.subversion/config ) files:

```
### Set merge-tool-cmd to the command used to invoke your external
### merging tool of choice. Subversion will pass 5 arguments to
### the specified command: base theirs mine merged wcfile
# merge-tool-cmd = merge_command
merge-tool-cmd = /home/xavi/scripts/svn-merge-meld
```

Once you saved your edit in the left pane (according tot he config explained in the example at the previous url), you can exit Meld, and if you are happy with your changes, mark that conflict as resolved typing the letter "**r**".

And the script is:

**contents of svn-merge-meld**

```
#!/usr/bin/env python
# svn merge-tool python wrapper for meld
# Taken from http://codingundertheinfluence.blogspot.com.es/2010/01/using-meld-as-your-external-diff3.html
import os, sys
import subprocess
import shutil

try:
   # path to meld
   meld = "/usr/bin/meld"

   # file paths
   base   = sys.argv[-4]
   theirs = sys.argv[-3]
   mine   = sys.argv[-2]
   merged = sys.argv[-1]

   # Replace 'merged' file with a copy of 'mine'
   # (because we can't open 4 files)
   shutil.copy(mine, merged)

   # the call to meld
   # work right-to-left (result is far left file)
   cmd = [meld, merged, theirs, base]

   # Call meld, making sure it exits correctly
   subprocess.check_call(cmd)
except:
   print "Oh noes, an error!"
   sys.exit(-1)
```

Alias names for this page:

subversion | svn

[1] http://codingundertheinfluence.blogspot.com.es/2010/01/using-meld-as-your-external-diff3.html