Semi-automatic merging period

> Please see Where to Commit in addition to this page

As per Version lifecycle, Tiki has two major releases per year.

Thus, there is the creation of two new branches per year. For a 2 to 3 month period after branching, there is a stabilization period during which there is a merging from branch to trunk using a script. When to branch is decided by the Release Coordinator

This period typically ends when merging becomes too difficult because the code in trunk starts to differ significantly. This is usually the time to proceed to the x.1 release and for the Quality Team to go in "strict mode".

During this period, it's best to avoid massive changes (refactoring, code cleanups, re-indenting. etc.) on trunk because it makes merging more difficult. If you want/need to do, please also do the merges.

All developers should upgrade their sites during this period as bug fixing is very efficient.

Merging from branch to trunk

1. Check-out trunk.
2. Using the command line, move to the root directory where you have the clean copy of trunk.
3. Make sure your version of trunk is up to date and there are no uncommitted changes (the script below will also check this).
4. Using the command line at the trunk directory, run `php doc/devtools/svnbranchupdate.php branches/version_number`
    - Example: `php doc/devtools/svnbranchupdate.php branches/20.x`
      This command is pointing to the branch on the repository, not your copy.
5. Run `sh setup.sh` and click `c` when prompted. This will update composer.
6. Fix conflicts and commit using `svn commit -F svn-commit.tmp`
    - If the merge changes **composer.json** , it is likely that **composer.lock** will have a conflict during the merge.
      To resolve this:
        1. Use `mc` (mine conflict)
        2. Run `php temp/composer.phar update` *(without the --no-dev option)*
        3. Add a comment to the **svn-commit.tmp** indicating the conflict you fixed.
        4. Commit as normal.
           See example below.
7. If there are conflicts that you cannot fix, try `svn revert -R ./` to reverse the merge and notify the dev mailing list.
    - Whatever you do, please don't just commit parts of the changes as that results in changes getting missed between the versions.

If we are outside this period, you can manually merge a commit.

Examples

```
[demotw@alpha trunk]$ php doc/devtools/svnbranchupdate.php branches/15.x
Verifying...
Updating...
Merging...
--- Merging r35997 through r36007 into '.':
U    lib/wiki-plugins/wikiplugin_addtogooglecal.php
After verifications, commit using `svn ci -F svn-commit.tmp`
```

Example of composer conflict

[+]

Q & A

Q: How do I check there is a conflict?

A: I don't think there's a way to (easily) check for conflicts, but when you run the svnbranchupdate.php script it only affects your local copy and it will say if there are conflicts, so nothing will get committed unless doing the "svn commit" part (and I believe it will refuse to commit while conflicts are unresolved.

- Another tip is that if you do get conflicts and either don't have the time or feel unable committing them you can revert the changes on your local trunk working copy (and call for help!). The command is:

```
svn revert -R .
```

Q: How to resolve conflicts?

You will be notified of any conflicts that arise from merging and offered a number of choices for how to resolve. See the tutorial below for an explanation of the choices (even if you're not using PhpStorm).

Note that the *mine-conflict* choice (or *mc*) means that the code in trunk will be used, whereas *theirs-conflict* (or *tc*) means that the code in the branch will be used to replace what is in trunk.

Using PhpStorm

Below is an excellent tutorial by Jonny Bradley on resolving merge conflicts using PhpStorm:

Using Meld

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

Once you saved your edit in the left pane (according to the 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**".

Q: How to resolve a tree conflict?

Sometimes, a file is added to the file tree in trunk first, and later, that file is added and backported to an earlier branch. That is usually a source of a tree conflict when you attempt to run the svnbranchupdate.php script.

For instance, if the file in conflict is img/icons/large/admin_panel48x48.png , you'll get something like:

```
user@server:/var/www/trunk_clean# svn ci -F svn-commit.tmp
svn: E155015: Commit failed (details follow):
svn: E155015: Aborting commit: '/var/www/trunk_clean/img/icons/large/admin_panel48x48.png'
remains in conflict
```

Some workaround to allow you to commit your other changes, is to revert (according to svn) the changes to that file, with:

```
root@coprinus:/var/www/trunk_clean# svn revert img/icons/large/admin_panel48x48.png
Reverted 'img/icons/large/admin_panel48x48.png'
root@coprinus:/var/www/trunk_clean# svn ci -F svn-commit.tmp
Sending        .
Sending        lib/core/Math/Formula/Function/Avg.php
Sending        lib/core/Math/Formula/Function/SplitList.php
Sending        lib/jquery_tiki/tiki-maps.js
Sending        lib/modules/modlib.php
```

```
Transmitting file data ....
Committed revision 49428.
```

That conflict could also be resolved, probably, with:

```
svn resolve --accept working img/icons/large/admin_panel48x48.png
```

Related

- Mass operations to do after the Semi-automatic merging period

alias

- Semi-automatic merge period