

SvnMerge

Subversion Merge

- Subversion Merge
 - Why merge?
 - svn:mergeinfo
 - Full Merge
 - Subversion 1.8 Note
 - Partial Merge
 - Merge Conflicts
 - Handling Moved Paths
 - example merging with moved path from /trunk to branch_3x
 - Example merge from trunk to branch_3x
 - References

Why merge?

Unlike a more simplistic approach of committing the same patch to both the trunk and the branch, using `svn merge` helps preserve file history information (like renaming files, or adding new files).

svn:mergeinfo

Subversion tracks merges using a property called `svn:mergeinfo`.

The problem: On `branch_3x/`, paths that were merged from `trunk/` *below the top level directory* began to accumulate revisions for every single commit that was the result of a merge, even when the commits did not include any direct changes to these paths. This is because a path with an `svn:mergeinfo` value that differs from its parent's has to keep its own private copy of *everything*, including a full copy of its parent's merge information. As a result, every time any of such a path's ancestors is a merge target, the private `svn:mergeinfo` value has to be updated and included in the `branch_3x/` commit.

The fix: Always run `svn merge` from the top level directory (i.e. the directory containing `solr/` and `lucene/`). There are two merge scenarios: a) fully merging all changes from a `trunk/` commit into `branch_3x/`; and b) performing a partial merge, where only some of the changes from the `trunk/` commit will be merged into `branch_3x/`.

Full Merge

To merge **all** of the changes from a `trunk/` commit into `branch_3x/`, go to the top level directory in your `branch_3x/` working copy, and issue this command:

```
svn merge -c XXXXXX https://svn.apache.org/repos/asf/lucene/dev/trunk
```

where `XXXXXX` is the revision of the source `trunk/` commit.

Subversion 1.8 Note

it seems like svn 1.8 resolves all these complications. After, you committed to trunk just change to branch working copy and run

```
svn merge ^/lucene/dev/trunk -cXXXXXX
```

Partial Merge

To merge a subset of changes from a `trunk/` commit into `branch_3x/`, go to the highest level directory possible containing the changes you want to include, but not containing any changes you want to exclude, and run `{{ svn merge -c XXXXXX }}` with the corresponding repository URL. Repeat until you have merged all of the desired changes.

Finally, go to the top level directory and issue the following command:

```
svn merge --record-only -c XXXXXX https://svn.apache.org/repos/asf/lucene/dev/trunk
```

The `--record-only` parameter instructs Subversion to skip performing any actual merging, but to instead modify the `svn:mergeinfo` property values to include the `XXXXXX` revision. Crucially, in addition to modifying the `svn:mergeinfo` property value for the top level directory itself, all paths' `svn:mergeinfo` property values are also adjusted, so that merged paths with the same merge information as their parents don't carry their own private `svn:mergeinfo` copies.

Merge Conflicts

If your `trunk/` commit contained changes to paths that are not in exactly the same place on both `trunk/` and `branch_3x/` (e.g. moved paths), Subversion will report a merge conflict.

Once you have merged the desired changes from the `trunk/` commit – see the instructions at [Partial Merge](#) – tell Subversion that you have resolved the conflict(s):

```
svn resolve --accept=working your/conflicted/path
```

Handling Moved Paths

The Subversion book [says](#):

For long-lived release branches (as described in [the section called "Common Branching Patterns"](#)), perform merges only on the root of the branch, not on subdirectories.

However, when one or more of the changed paths in the `trunk/` commit has been moved from its location under `branch_3x/`, e.g. `modules/analysis/*` and `modules/benchmark`, merging below the top level directory is necessary; follow the instructions under [Partial Merge](#) and [Merge Conflicts](#).

Without taking further action, though, each merged moved path will carry its own private `svn:mergeinfo` copy of all its ancestors' merge information (see [the `svn:mergeinfo` section](#) for an explanation of why this is a problem).

For the specific case of merged moved paths, to avoid the `svn:mergeinfo` property value accumulation problem on `branch_3x/`, we remove the `svn:mergeinfo` property from all paths except for the top level directory and its immediate children `solr/` and `lucene/`.

`svn stat` indicates the paths with modified properties via an

```
"M"
```

in the second status column from the left. For each of these paths (except for the top level directory, `solr/`, and `lucene/`), run the following command:

```
svn propdel svn:mergeinfo remove/mergeinfo/from/this/path
```

example merging with moved path from /trunk to branch_3x

```
you@machine:~/workspace/lucene-branch3x/lucene/contrib/icu$ svn merge -c 1130547 ~/workspace/lucene-clean-trunk/modules/analysis/icu
--- Merging r1130547 into '.':
U build.xml
you@machine:~/workspace/lucene-branch3x$ svn merge -c 1130547 ~/workspace/lucene-clean-trunk
--- Merging r1130547 into '.':
C modules
Summary of conflicts:
Tree conflicts: 1
you@machine:~/workspace/lucene-branch3x$ svn resolved modules
Resolved conflicted state of 'modules'
you@machine:~/workspace/lucene-branch3x$ svn propdel svn:mergeinfo lucene/contrib/icu
property 'svn:mergeinfo' deleted from 'lucene/contrib/icu'.
you@machine:~/workspace/lucene-branch3x$ svn commit -m "merge r1130547"
```

Example merge from trunk to branch_3x

1. `svn merge https://svn.apache.org/repos/asf/lucene/...` at the top level, noting tree conflicts, e.g. from `modules/`
2. For each of the conflicts, `svn resolve --accept=working conflicted/path/here`
3. For each changed path/file, figure out the highest possible path (root is highest) in common between `trunk` and `branch_3x`, change to that directory in `branch_3x`, then merge from the corresponding trunk dir.

4. `svn stat` to discover which directories and files have accumulated `svn:mergeinfo` properties below the top-level directories that get generated by merging below the top-level dir - the second column from the left for each changed item will have an

```
"M"
```

for property changes

5. For each of the dirs/files below the top-level dir (except `solr/` and `lucene/`, which are allowed to accumulate `svn:mergeinfo`), `svn propdel svn:mergeinfo path/to/item/with/mergeinfo`
6. `svn stat` again to be sure nothing is left out or unintentionally included
7. Compile/test, if the build is affected by the changes
8. `svn commit`

References

- From the Subversion book:
 1. The `{{ svn merge }}` command: <http://svnbook.red-bean.com/en/1.5/svn.ref.svn.c.merge.html>
 2. Basic Merging: <http://svnbook.red-bean.com/en/1.5/svn.branchmerge.basicmerging.html>
 3. The final word on advanced merging: <http://svnbook.red-bean.com/en/1.5/svn.branchmerge.advanced.html#svn.branchmerge.advanced.finalword>
- The gory details of Subversion's `{{ svn:mergeinfo }}` property handling: <http://www.collab.net/community/subversion/articles/merge-info.html>