



The Future of Merging in Subversion



Julian Foad

June 2010

Julian Foad

▶ **WANdisco**

Subversion: obliterate, WC-NG

- ***PART I – NOW***
 - *“A better CVS”*
 - *Customer feedback*
 - *Other VCS*
 - *Working on WC-NG for 1.7*
- ***PART II – NEXT***
 - *Speed*
 - *Renames*
 - *Conflicts*
 - *Work Flow*

- *PART I - NOW*
 - *“A better CVS”*
 - *Customer feedback*
 - *Other VCS*
 - *Working on WC-NG for 1.7*
- *PART II – NEXT*
 - *...*

- ▶ 2000 – We wanted “a better CVS” ...
- ▶ 2003 ... and got it.
- ▶ 2010 – Much better, but ...
- ▶ 2013 ... what do we want?

- ▶ Merging is a big issue
 - “merging is the act of collaboration”
 - <<http://www.markshuttleworth.com/archives/126>>
 - Speed, Renames, Quality

▶ Major customers consulted

- By C.M.Pilato (CollabNet), H.Wright (WANdisco)
- See *<notes/feedback/*>*

▶ Results

- **Speed**
- Authentication and authorization
- Server-side control of client-side behaviors
- Improved **branching and merging**
 - **Conflicts, Renames**

- ▶ Merging
- ▶
- ▶ * More **robust** merging
- ▶ * --reintegration should tolerate **sparse checkouts** that aren't affected
- ▶ * Use common ancestor as part of the merge resolution process
- ▶ * First class citizenship for mergeinfo (revision- and path-level support)
- ▶ * Better handling of **property merges** (particularly svn:mergeinfo)
- ▶ * Support path-level merge **traceability** without adding heavy overhead
- ▶ * Make record-only merges support transitive merges
- ▶ * Support cyclic merging (**multi-reintegrate** support)
- ▶ * Support --changelist option on 'svn merge' (limit merge targets)
- ▶ * Avoid creating svn:mergeinfo when merging into a sparse working copy
- ▶

- ▶ Speed
- ▶ Renames
- ▶ Conflict resolution
- ▶ Work flow
 - re-integrate limitations, cyclic merging

- ▶ **What should Subversion aspire to?**
 - Git's speed
 - Bazaar's rename handling
 - Subversion's centralized model

- ▶ Working on WC-NG for v1.7
- ▶ For merging, this means
 - Renames can be tracked in the local WC
 - Many client-side operations much faster

- *PART I – NOW*
 - ...
- *PART II – NEXT*
 - *Speed*
 - *Renames*
 - *Conflicts*
 - *Work Flow*

- ▶ **Speed means...**
 - Do everything possible to help the user

- ▶ **Catch-up merge (re-basing a branch)**
 - Often, many files change, but few need merging
 - Should be as fast as “update”

- ▶ **Re-integrate merge**
 - After the checks, it is like a “copy” operation
 - Should be as fast as “update”

- ▶ **WC-NG**
 - Client side speed. Effect on merge speed not yet known.

- ▶ Renaming needs to be Easy
- ▶ Renaming needs to be Reliable
- ▶ Rename Tracking means ...
- ▶ Steps needed
 - Client side
 - Protocols
 - Repository side
- ▶ What's possible now / Short cuts?
 - ...?

```
$> svn rename Dir Dir2
```

```
$> svn status
```

```
V      Dir2 (from Dir)
```

```
$> svn commit
```

```
Renaming Dir to Dir2
```

```
Committed revision 22.
```

```
$> svn add Dir/newfile
```

```
A      Dir/newfile
```

```
$> svn update
```

```
V      Dir2 (from Dir)
```

```
$> svn status
```

```
A      Dir2/newfile
```

```
$> svn merge ...  
V      Dir2 (from Dir)
```

```
$> svn status  
A      Dir2/newfile
```

- ▶ Conflict detection and resolution
- ▶ Better non-interactive conflict resolution
 - Improvements coming in 1.7

- ▶ **Re-integrate merges**
 - ▶ Should tolerate sparse checkouts that aren't affected
 - coming in 1.7
 - ▶ Should not be necessary to use a Working Copy – just supply a log message and have it merge and commit in one step.
 - ▶ Should support continuing work on the same branch and then re-integrating again.
- ▶ **Transitive merges (A > B > C > B > A)**

▶ Analyze merge work flows

- Merge algorithms
- Expectations
- What can we do already?

- ▶ Current state
- ▶ WC-NG
- ▶ Hg, Git, etc.
- ▶ Aiming for?

Merging

Any questions?