Shelving-v2 in Svn-1.11

A summary of the experimental Shelving and Checkpointing feature in Svn-1.11, building on the initial Shelving-v1 in Svn-1.10.

(For other pages about Shelving and Checkpointing, see Shelving and Checkpointing.)

Summary

This version of shelving (issue SVN-3625) builds on the initial Shelving-v1 in Svn-1.10.

It adds support for making checkpoints and rolling back to an earlier checkpoint (issue SVN-3626).

The implementation is no longer based on patch files. Instead, the base and working version of each file are stored when shelving. To apply the changes, 3-way merge is used for text files, and theirs-or-mine-or-conflict logic for binary files, similar to the way "svn merge" works.

The kinds of change that can be shelved and unshelved are detailed below.

Functionality

Shelve and Unshelve

A shelf is created from WC changes based on a particular repository, branch and revision, but is not tied to those details. It is intended that a shelf can be unshelved (applied) on to any WC target that is similar to the original base, be it another branch, another revision, or (future possibility) another repository, similar to how patch files can be used.

Shelves are stored in the metadata area of a working copy, and used within that WC.

- to unshelve into a different revision: 'update' the WC to the new revision, then unshelve
- to unshelve into a different branch: 'switch' the WC to the new branch, then unshelve
- FUTURE: move a shelf to a different WC
- FUTURE: specify a different target WC when shelving/unshelving
- FUTURE: store shelves in a central location, such as user's home directory

Save a Checkpoint and Restore (Roll Back) to a Checkpoint

Checkpoints are supported by saving multiple versions of a shelf.

Save a checkpoint	`svn shelf-save foo`	copy the local changes into a new version of shelf 'foo' **doesn't revert** the changes from the WC
	`svn shelve foo`	move the local changes into a new version of shelf 'foo' **and revert** the changes from the WC
Restore / roll back	`svn unshelve foo 3`	unshelve version 3 of shelf 'foo' **and delete any newer versions**
`Review checkpoints`	`svn shelf-log foo`	list all the versions of shelf 'foo'
	`svn shelf-diff foo 3`	show version 3 as a diff

Command-Line UI

$ svn help

```
Main commands:
 x-shelf-drop (shelf-drop)  # delete a shelf
 x-shelf-list (shelf-list, shelves)  # list all shelves
 x-shelf-log (shelf-log)    # list the versions of a shelf
 x-shelf-save (shelf-save)  # save, don't revert
 x-shelve (shelve)          # save and revert
 x-unshelve (unshelve)      # unshelve or restore

Commands under development:
 x-shelf-diff (shelf-diff)
 x-shelf-list-by-paths (shelf-list-by-paths)
```

$ svn help shelve

```
x-shelve (shelve): Move local changes onto a shelf.
usage: x-shelve [--keep-local] SHELF [PATH...]

  Save the local changes in the given PATHs to a new or existing SHELF.
  Revert those changes from the WC unless '--keep-local' is given.
  The shelf's log message can be set with -m, -F, etc.

  'svn shelve --keep-local' is the same as 'svn shelf-save'.

  The kinds of change you can shelve are committable changes to files and
  properties, except the following kinds which are not yet supported:
     * copies and moves
     * mkdir and rmdir
  Uncommittable states such as conflicts, unversioned and missing cannot
  be shelved.

  To bring back shelved changes, use 'svn unshelve SHELF'.

  Shelves are stored in <WC>/.svn/shelves/
```

$ svn help x-unshelve

```
x-unshelve (unshelve): Copy shelved changes back into the WC.
usage: x-unshelve [--drop] [SHELF [VERSION]]

  Apply the changes stored in SHELF to the working copy.
  SHELF defaults to the newest shelf.

  Apply the newest version of the shelf, by default. If VERSION is
  specified, apply that version and discard all versions newer than that.
  In any case, retain the unshelved version and versions older than that
  (unless --drop is specified).

  With --drop, delete the entire shelf (like 'svn shelf-drop') after
  successfully unshelving with no conflicts.

  The working files involved should be in a clean, unmodified state
  before using this command. To roll back to an older version of the
  shelf, first ensure any current working changes are removed, such as
  by shelving or reverting them, and then unshelve the desired version.

  Unshelve normally refuses to apply any changes if any path involved is
  already modified (or has any other abnormal status) in the WC. With
  --force, it does not check and may error out and/or produce partial or
  unexpected results.
```

Kinds of change that can be shelved

WC State or Change	Shelving v1	Shelving (in development)	Planned
committable changes			
file text, file delete/add, most properties	yes	yes	yes
mergeinfo changes	yes	yes	yes
copies and moves	no	no	yes
directories (mkdir/rmdir/...)	no	no	yes
binary files & properties	no	yes	yes
uncommittable state			
unresolved conflicts	no	no	no

changelist assignment	no	no	no
unversioned items (git stash -u/-a)	no	no	no
inconsistent WC states (missing/obstructed)	no	no	no
base state			
WC base state (rev, URL, switched, depth)	no	no	?

Conflicts and Error Handling

When 'svn shelve' encounters any unshelvable WC state in the specified paths:

- error out, refusing to shelve any changes
  - error message gives some details of why it failed

When 'svn unshelve' hits a path that already has a local modification:

- error out; the error message gives some details of why it failed
- the shelf is not changed
- the WC is not changed
- with 'svn unshelve --force' option: bypass this check

(Why not unshelve with local modifications, by default? In the main use cases for unshelving, we expect that the user has a clean working copy. If the relevant files in the WC are not 'clean', it may be the user has made a mistake. And after we merge some shelved changes into the local modifications, if the user does not like the result, it is not possible to 'undo' the merge, in general. In some cases a reverse-merge may work but in general merging is not an automatically reversible process. But if the user is being careful, and deliberately wants to merge the shelved change with some local changes, we should make it possible to do this, so we have a 'force' option.)

When 'svn unshelve' hits a path that has different content from the base of the shelved change (for example, if the WC is switched to a different branch or updated to a different revision), it tries to merge:

- for text file content: 3-way merge
- for binary file content (as determined by 'svn:mime-type' property): theirs-or-mine-or-conflict
- properties: theirs-or-mine-or-conflict for each property

When 'unshelve' hits a conflict in merging file or properties:

- raise a conflict state in the WC for that path, just like 'merge' does.

Implementation

- library functions implemented in libsvn_client
- not yet exposed through bindings (SWIG, JavaHL, etc.)

Client-level support (user interfaces):

- 'svn' command-line UI
- TortoiseSVN (for Windows)
- Cornerstone (for Mac)

Shelves are stored in WC metadata dir '.svn/shelves/'.