

Practical Git for Smalltalk

STIC 2012

Dale Henrichs

VMware

Practical Git for Smalltalk?

It is PRACTICAL to use Git to manage Smalltalk source code.

It is IMPRACTICAL for all dialects to switch to wholesale use of Git.

Using Git for sharing code between Smalltalk dialects makes a lot of sense.

Why don't we all use Envy/Store/ Monticello?

We could, but there is a high barrier to entry:

- Proprietary storage formats
- High cost of porting and tool set integration
- Incomplete solutions
 - Git has some features no found in the Smalltalk SCMs

FileTree

<https://github.com/dalehenrich/filetree>

A simple, disk-based Package structure for storing class and method source (Snapshot) on disk:

- Originally developed by Otto Behrens
- Implementation is based on Monticello, to integrate with Squeak/Pharo/GLASS development environment
- Disk structure does not rely on any Monticello artifacts
- Use Package/Snapshot solves "naive .st filein" problem
- The FileTree Package structure is SCM neutral.
 - can be used with Svn, Mercurial, Git, etc.

What's a Package/Snapshot?

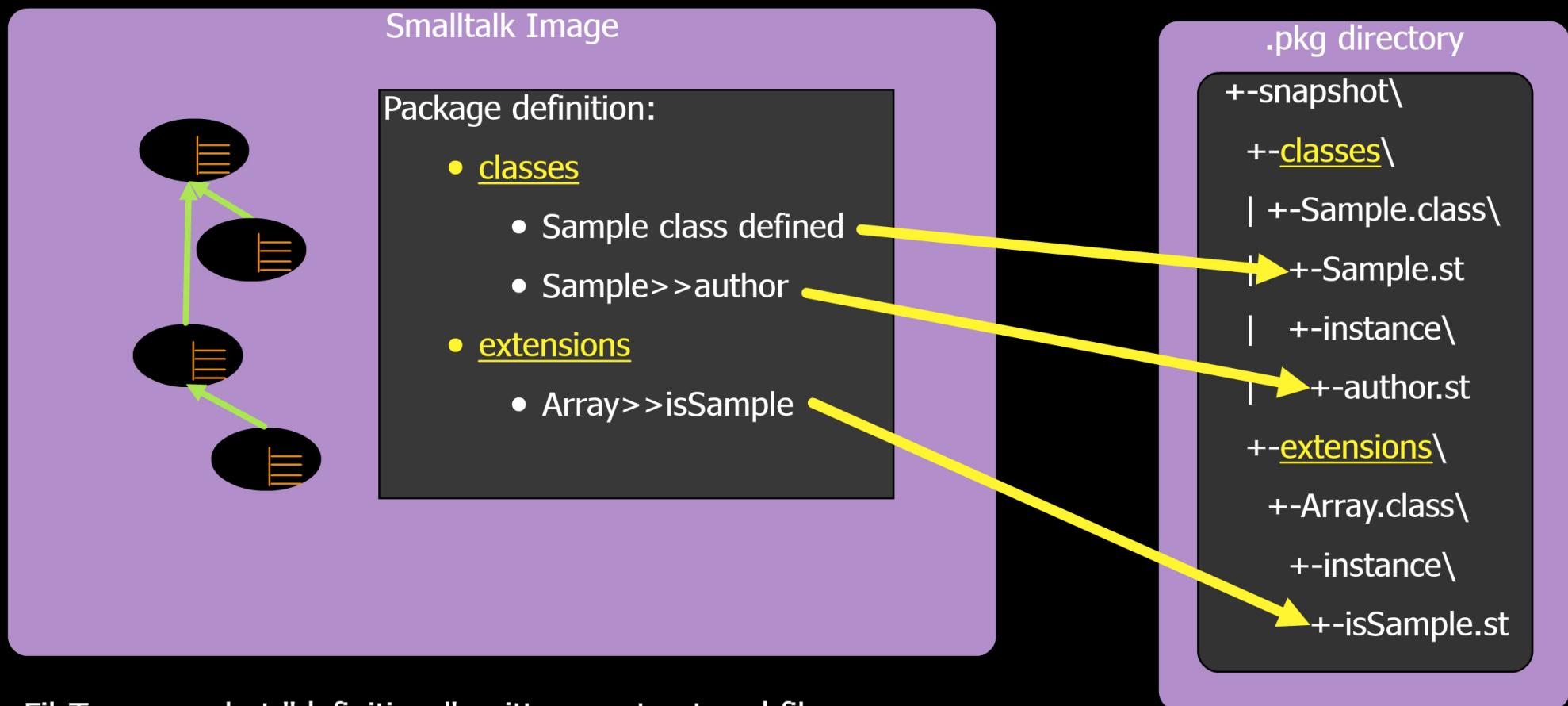
A Package is an object or algorithm that defines the contents of a Snapshot.

A Snapshot is the state of a Package at a point in time.

The Snapshot is represented in the image as a collection of class and method "definitions".

The Snapshot is represented on disk as class and method chunks in a collection of .st files.

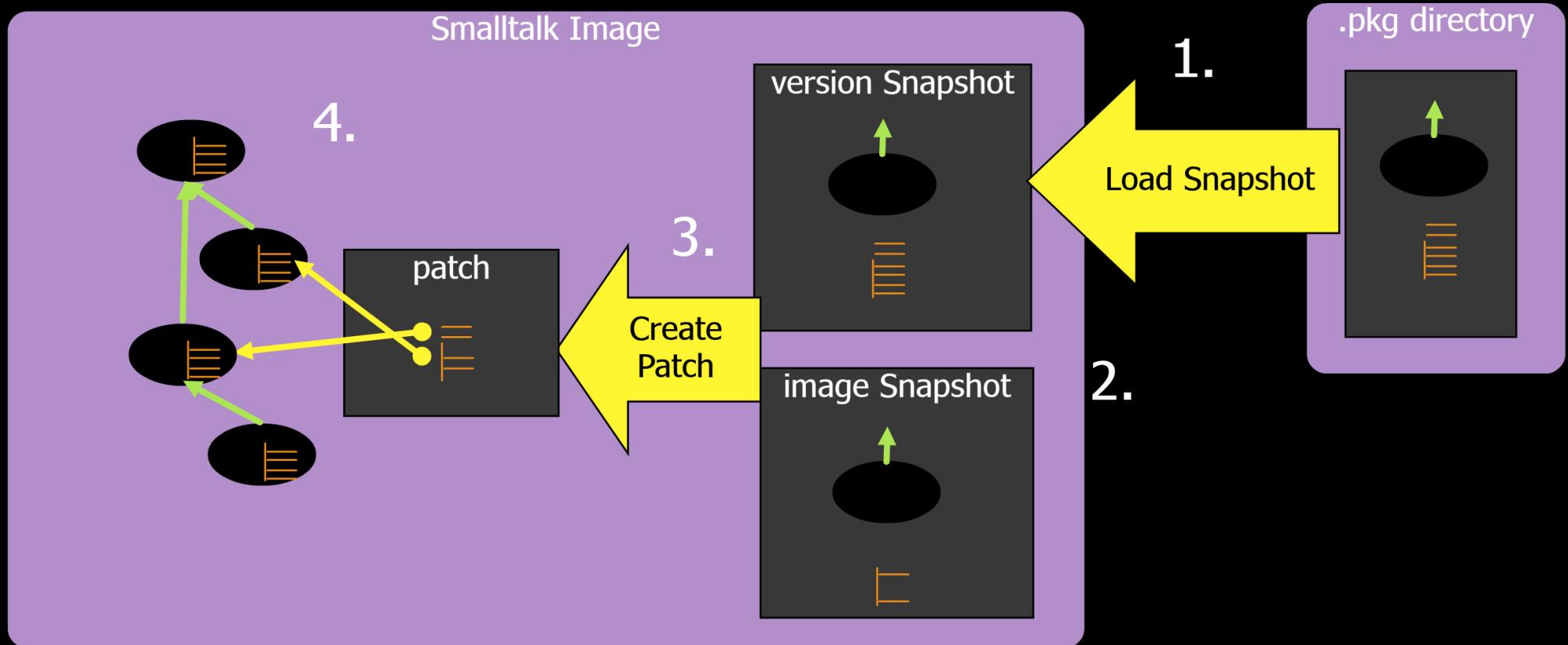
FileTree Package Structure



FileTree snapshot "definitions" written as structured files:

- classes directory for class and method definitions
- extensions directory for loose methods

FileTree Snapshot loading



1. Load version snapshot from .pkg directory
2. Create image snapshot of current image
3. Create Patch by comparing snapshots
4. Apply patch, adding/removing/changing classes and methods

FileTree Status

Functional for Monticello-based development environments:

- Likely to be tweaks as FileTree project is ported to other dialects
- Built-in flexibility accommodates multiple file structures
- Otto's been using FileTree in production for 5 months and I've used it for nearly 2 months

3 Phase FileTree Plan

Phase 1

- FileTree format stabilized for Phase 2 use
- Metacello scripting API with GitHub support in-progress

Phase 2

- Early adopters use Git/GitHub to develop usage guidelines
- Port FileTree support to other dialects

Phase 3

- add in-image support for Git common operations
- Finalize release plan

What is Git?

Disk-based, Distributed Version Control System for source code

- Each instance is a complete copy of repository
- "Clone" repository to local disk for off-line work
- Merge changes to/from any other clone

Branches are "first-class" objects

- Branches enforced by api

Designed for performance

- Always work from clone on local disk faster than network access

Free repository hosting for Git repositories

What is GitHub?

GitHub (<http://github.com>)

- Wiki/blog/issue tracking for each project
- Code review/integration support ("pull request")
- Branch visualization
- Source comparison
- Over 1,000,000 GitHub users

Why aren't we using Git/GitHub now?

Git is relatively new:

- Envy/Store (199x?)
- Monticello (2003)
- Git (2005)
- GitHub (2008)
- Metacello (2009)

In the 90's Envy/Store was superior to existing SCMs options, especially for Smalltalk.

Git and GitHub have leap-frogged current options.

Why are you pushing Git/GitHub?

Compelling feature set.

GST, Redline and Amber host their projects on GitHub.

Have to pick one to start things off.

I'm happy with Monticello, why should I use Git?

Git provides superior functionality:

- Monticello versions a single file, while Git versions an entire directory structure
- save non-Smalltalk source artifacts (html, css, icons, etc.) in repository
- take advantage of Git branching

With Metacello scripting API, you will be able to mix and match GitHub-based projects and traditional .mcz-based projects

I'm happy with Store/Env, why should I use Git?

Continue to use Store/Env as primary SCM.

Use Git/Github for sharing code with other Smalltalk dialects:

- only need to import/export "packages"
- tool support nice but not essential

I'm excited!

That Git and GitHub will provide a big lift for the Monticello-based development environments:

- The larger projects are hitting walls
- FileTree repositories make it trivial to migrate a project from using .mcz files to FileTree and Git
- New Metacello scripting API will simplify the scripting and loading process

I'm excited!

That Git and GitHub could finally facilitate true code sharing between dialects:

- Missing capability for way too many years (decades?)
- We are too small to afford the luxury of isolation.
- We must be able to find a way to combine our resources and collaborate on cross-dialect projects

Questions?

Resources

General FileTree and Git/Github discussion list:

- <http://groups.google.com/group/metacello>

FileTree GitHub project:

- <https://github.com/dalehenrich/filetree>

FileTree Issue list for implementation/feature discussions:

- <https://github.com/dalehenrich/filetree/issues>

FileTree bootstrap instructions (Pharo/Squeak):

- <https://github.com/dalehenrich/filetree/blob/master/README.md>