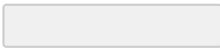


# Move from Subversion to Git

Tracked under



SLING-3987 - Move from Subversion to Git

CLOSED

## Overview

We will split current SVN repository located at <https://svn.apache.org/repos/asf/sling/trunk/> into multiple git repositories hosted on the ASF git servers. We will opt in to the Git "Dual Master" system which will allow us to push directly to Github. Direct access to Github and the possibility of directly merging pull requests can be a booster for community participation.

We will not enable other Github features, such as wikis, features, and projects.

The rule of the thumb for selecting what ends up in a git repository is a releasable Maven unit. The naming of git repositories will be `sling-${sanitizedArtifactId}`. The sanitized artifactId is composed from the Maven artifact id where all dots are replaced by slashed. See



INFRA-15229 - Jira project doesn't exist or you don't have permission to view it.

for details. The sling prefix is required as we will

share the [Apache Github org with other ASF projects](#).

## Commit history

Commit history will be preserved as we will convert the projects using git-svn, followed by git filter-branch. However, this has limitations when the project has been moved across multiple directories. We have detected it so far for the repoint modules. We consider this acceptable, as the full history also resides in SVN. There are no confirmed alternatives that can produce better results. Trying to import the whole ASF SVN repository to experiment with the reposurgeon tool is taking too much time - still not done after five days.

## Special situations

The following directories will not be split to individual modules:

1. tooling/ide - the modules are not individually buildable and releasable
2. performance - the modules are tied together in a performance testing harness
3. samples - having samples spread over multiple repositories will make it harder for users to discover all of them
4. whiteboard - since we intend to perform no releases from the whiteboard, it does not make sense to create individual repositories. Also, personal Github repositories can be used as well, given that the owner is an Apache committer and only they contribute to that repository.

We will not migrate the attic as there will be no further work done on the modules in there.

The following non-Maven directories will be migrated as separate repositories

1. tooling/scm
2. tooling/jenkins

## Unified view of the repository

We will create the special 'repo' sling which will contain one or more `repo` manifests which will allow viewing and editing multiple repositories at once.

## Canonical repository URLs

As part of the gitbox setup, the repositories can be accessed both via GitHub and [gitbox.apache.org](https://gitbox.apache.org). Since there is a strong requirement to cut releases from ASF servers, all pom references will point to gitbox instead of GitHub. Of course, committers are free to use GitHub as a remote for pushing, just releases are affected.

## Migrating a module

If an individual repository needs to be migrated, usually because it was missed from the initial migration, the following steps are needed (assuming we need to migrate `contrib/scripting/esx`):

```
$ git clone https://github.com/apache/sling-old-svn-mirror # clone the old svn which will be used as a source
for the migration
$ cd sling-old-svn-mirror
$ echo contrib/scripting/esx > repo-candidates.txt # file name is not important, but it's helpful to have the
name(s) stores in a central place
$ ./tooling/scm/scripts/migrate-to-git.sh -c < repo-candidates.txt # converts the repositories locally in ../
sling-modules/
$ ./tooling/scm/scripts/migrate-to-git.sh -r < repo-candidates.txt # provisions the git repository on github
/gitbox . Github permissions take 30-60 minutes to sync, you will not be able to push to this repo immediately
$ cp LICENSE ../sling-modules/sling-org-apache-sling-scripting-esx/ && cd ../sling-modules/sling-org-apache-
sling-scripting-esx/ && git add LICENSE && git commit -m 'Added LICENSE file' && cd -
$ ./tooling/scm/scripts/migrate-to-git.sh -p < repo-candidates.txt # after verifying that the repository is
writable on github
```

Once the repository is provisioned add it to the [sling-aggregator repo](#) by following the instructions from the aggregator README.

## Older information

### Tools and scripts

Some WIP tools can be found at <https://svn.apache.org/repos/asf/sling/trunk/tooling/scm/scripts/> .

### Can Sling (committers and contributors) benefit from moving to Git?

#### Pros

- offline support
- cherry picking
- easy to *move* projects (e.g. between bundles, contrib, whiteboard...)
- adopting the Github workflow using the Git Dual Master system
- ...

#### Cons

- not eating our own dog food
- slow for big repositories (many modules)
- tagging is global
- ...

### What needs to be prepared for moving to Git?

- repository layout
- release process
- documentation
- CI
- ...

### Git at Apache

<http://wiki.apache.org/general/GitAtApache>  
<http://www.apache.org/dev/git.html>

### Repository Layout and Projects (Modules)

There are currently 250 projects (modules) in our repository, counted using the `tooling/scm/scripts/gen-repo-candidates.sh` script ([repo-candidates.txt](#)).

Currently we consider every Maven project for extraction into its own repository. As exception we have:

- `tooling/ide`, which is a releasable unit by itself and not buildable as individual modules due to toolchain limitations
- `tooling/scm`, which contains various utility scripts
- `tooling/jenkins`, which contains the Groovy scripts used to generate the Jenkins jobs

Additionally, we will probably have one aggregator repo used to generate the full view over the git repositories, using `repo`, `gitslave`, or another tool that we will settle on.

### Idea

- one Git repository per project/module (we should keep it that simple - is launchpad really different, a release unit, [Justin Edelson?](#))

- grouping repositories with Google [repo](#), e.g. core (bundles), launchpad, testing, samples, tooling as needed
- using `artifactId` as repository name for artifacts (bundles, plugins, jars, ...) and simple names for *grouping* (builder/reactor) repositories
- using Google repo allows developers to set up their own grouping repositories - even non Sling committers

## Grouping repositories (projects/modules)

There are some tools for grouping repositories (projects/modules):

- [Google repo](#)
- [GitSlave](#)
- [git-subtree](#)
- [git-submodule](#)
- [git-subrepo](#)
- [Git External](#)
- [tsrc](#)

Some posts comparing different tools:

- <http://blogs.atlassian.com/2013/05/alternatives-to-git-submodule-git-subtree/>
- <https://codingkilledthecat.wordpress.com/2012/04/28/why-your-company-shouldnt-use-git-submodules/>

## References

dev@discussions:

- [\[jira\] \[Created\] \(SLING-3987\) move from Subversion to Git](#)
- [moving sling to git](#)
- [\[git\] Which git system do we want to use?](#)
- [\[git\] Please test sample migrated git projects and repo setup](#)
- [\[git\] Whiteboard migration](#)