

Git plugin 2.0 beta testing

- What's new in 2.0?
 - Simplified UI
 - JGit support
- Installation Procedures
- Activating JGit
- Reporting issues
- Hacking code
- Open Issues

We are doing a public open beta testing on the next major version of the Git plugin, and looking for people who can kick tires.

What's new in 2.0?

Simplified UI

In this release, we eliminated the entire "advanced" section and some of the default configuration items, and replaced them by a ala-carte menu of "additional behaviours" you can select and add from a drop-down:

The screenshot shows the 'Source Code Management' configuration page for the Jenkins Git plugin. The 'Git' radio button is selected. The 'Repository URL' field contains 'git://github.com/jenkinsci/git-plugin.git'. The 'Branches to build' section has a 'Branch Specifier (blank for default)' field with '**'. The 'Repository browser' is set to '(Auto)'. The 'Additional Behaviours' section has an 'Add' dropdown menu open, showing a list of options: 'Advanced clone behaviours', 'Advanced sub-modules behaviours', 'Check out to a sub-directory', 'Check out to specific local branch', 'Clean after checkout', 'Create a tag for every build', 'Credential for authentication (JGit only/experimental)', 'Custom SCM name', 'Custom user name/e-mail address', 'Don't trigger a build on commit notifications', 'Fast remote polling', 'Merge before build', 'Polling ignores commits from certain users', 'Polling ignores commits in certain paths', 'Prune stale remote-tracking branches', 'Strategy for choosing what to build', 'Use commit author in changelog', and 'Wipe out repository & force clone'. The 'Build Triggers' section has four unchecked checkboxes: 'Build after other projects are built', 'Build periodically', 'Build when another project is built', and 'Poll SCM'. The 'Build' section has an 'Add build step' dropdown menu.

This makes it easier to see the active configuration of the current job, and make Git plugin more open for other plugins.

JGit support

This release brings JGit as an option to work with repositories. As JGit runs inside Jenkins JVM and offers more finer-grained control, it brings a number of benefits difficult to achieve with native Git implementation, such as:

- Controlling authentication on per-project basis without prior setup on slaves
- No need to have native Git installed on slaves.

Beware of some known limitations with JGit:

- Symlink support

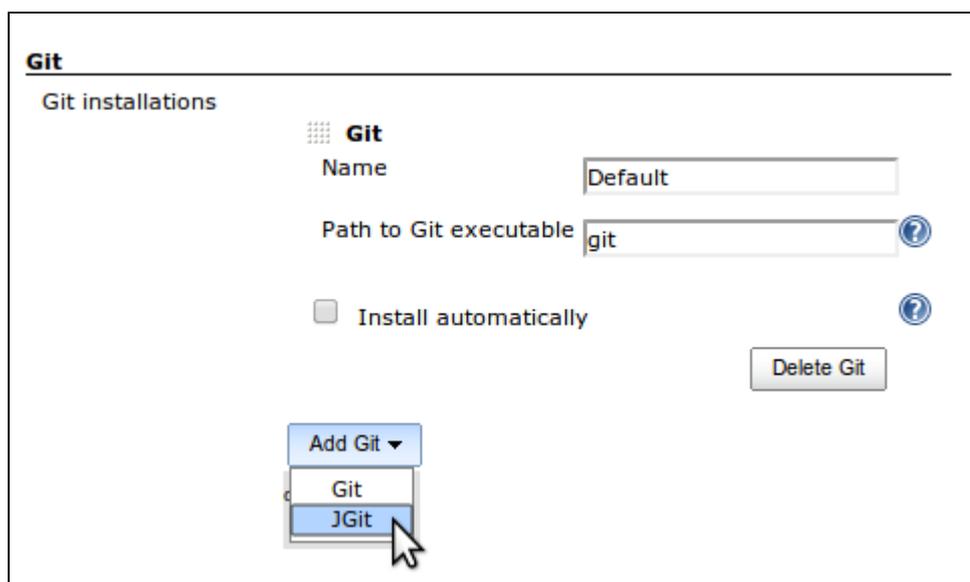
Installation Procedures

- Set your update center to the experimental update center
- Use plugin manager to install Git plugin 2.0 beta.
- Restart Jenkins, go back to the plugin manager and make sure you see the 2.0-beta version

Activating JGit

One of the highlights of Git plugin 2.0 is the use of JGit for working with Git repositories. This mode has some additional features, such as being able to set authentication credential inside Jenkins, instead of doing so on every slave.

To use JGit, go to the system configuration page and create JGit as the Git installation. If you remove whatever existing Git installation you might have configured, all your jobs will start using JGit (or if you leave existing git configurations in there, then each job can select whether it uses native git clients or JGit.)



Reporting issues

If you have problems with the beta versions of Git plugin, please report them on <https://issues.jenkins-ci.org/> When you report bugs, please add "git2" as a label to help us find them more easily.

Hacking code

If you want to hack the code, it is [here](#). It's in the "refactoring" branch.

Open Issues

Type	Status	Key	Summary	Reporter	Assignee
	Can't show details. Ask your admin to whitelist this Jira URL.				
			View these issues in Jira		

