

inheritance-plugin

Plugin Information

View Project Inheritance [on the plugin site](#) for more information.

The purpose of this plugin is to bring true inheritance of properties between multiple job definitions to Jenkins. This allows you to define common properties only once and inherit them to multiple projects.

Description

The role of this plug-in can be broadly described as bringing the concept of **property inheritance** to Jenkins. The goal is to fulfil the following statements:

- *Instead of having to define the same property multiple times across as many projects; it should be possible for many projects to refer to the same property that is defined only once.*
or rephrased:
- *Everything that is defined multiple times but used in the same way, should be defined only once and simply referred to many times.*

In a way, you can think of this plugin this way: **If Jenkins is the C language, this plugin is its C++ equivalent.**

Ideal candidates for this are, for example, the SCM setup, common pre-build and post-build steps as well as cleanup and logging properties.

Additionally, the plugin offers the following nice feature additions:

- A simple versioning system, allowing you to easily go back to a past version of any job configuration and test new features without affecting the runtime.
- The ability to use a custom workspace location (that can optionally make use of parameter values) that uses Jenkins' workspace locking mechanism so that the Job can still be executed multiple times on the same host without collision.
 - This is highly useful if many projects make use of the same remote repository. Just use the repository name as workspace location to cut down on the number of workspaces.
- The ability to automatically generate child jobs by joining multiple parent jobs together according to a parameterized "recipe".
 - *Do note* that this ability is currently too inflexible for broad use. It will be made much easier to use in future versions, though.
- Possibility to define a special "magic" label that makes sure that only those jobs that depend on this label get executed on a host. Think of it this way:
 - Normal Jenkins labels are useful to make sure that a particular set of jobs can run on a particular set of machines.
 - The "magic" node label makes sure that **only** a particular set of jobs can run on a particular set of machines -- and no others.
 - More precisely, it makes sure that those jobs that **do not reference the label** will **not** be executed on those hosts that **have this label**.

Video tutorials

The following video tutorials are currently available:

- General introduction on how to use the major features of this plugin:
<http://youtu.be/DNUI7DRI6XE>

Known Issues:

There are the known issues for version 2.0.0.

- JaCoCo plugin
 - The JaCoCo build trend graph does not render correctly, if the JaCoCo property is inherited, instead of defined on the leaf projects. The coverage report shows up, but the graph will fail to load properly. This can lead to all HTTP threads of Jenkins to become busy with rendering the graph, until the action times out and fails (after 1-2 minutes).

Changelog

Version 2.0.0 (14. March 2017)

This commit combines all the changes that we implemented in 2015 on GitHub (without a release) and in the past 2 years for our internal version. Since this is a dramatic and large change, the version was set to 2.0.0.

The code should be compatible with servers that were running the last version 1.5.3 (as long as they're running Jenkins > v1.625.3), but no guarantees can be made, as we can't check all edge-cases.

As such **extreme care** should be taken when upgrading from 1.5.3.

The most relevant changes are:

- The plugin now handles the "full name" of projects correctly. This enables compatibility with the CloudBees "Folders" plugin.

- Improved reporting of cycles and missing parameters in the inheritance. For example, the warning now lists the exact path of the cycle.
- Child jobs are now properly marked as unbuildable when there's a missing parameter or parent job, even when there are many levels of inheritance between it and the missing job/parameter.
- Job configuration "save" button will now check for errors (red text) in the configuration form and prevent you from saving if errors are present. This avoids errors on the page from creating half-changed jobs.
- The "Apply" button has been removed, as it does not allow giving a text for the new version that is created.
- When a build is started with a different version, a build description is automatically added.
- Fixed various issues relating to rebuilding existing jobs.
Examples: Wrong versions, missing parameters, etc. pp.
- Fixed leaks in the version handling, that sometimes caused builds to be started with versions selected for a wholly different run, or with a more recent unstable version, even if a stable version was selected.
- Improved layout of various overview pages:
 - Inheritance relationships,
 - Versioning and diff tables
 - 'Full build flow' for parameters and shells executed by a build
 - Parameter derivation page
- Made version comparison page hot-linkable: `!showDiffOfVersions?left=<num>&right=<num>`
- The `"/job/<name>/config.xml"` now accepts a `"?version=<num>"` parameter and will also, by default, show the XML of the last stable version, instead of the last-saved version.
- Creating a new job as a copy of an existing job will now also use the version currently marked as stable, instead of the last saved one.
- Inheritable Parameter References no longer store unused fields in the XML that only belong to "real" parameters.
- Inheritable Parameters now have full control over how Whitespaces are handled (i.e. are they kept, removed or automatically added?)
- Parameters are now aware of the inheritance as it is defined on the current configuration page. It is no longer necessary to save the project after changing the parentage to use "parameter references".
- Made "rebuild" plugin mandatory, as not having it led to confusing exceptions during boot.
- Fixed various exceptions raised in the GUI as well as backend.
- Removed a possible deadlock when deleting a job while saving another.

Version 1.5.3 (29. July 2014)

- Identical to v1.5.2.
Maven botched the update process of 1.5.2. It missed uploading a commit on Github and uploaded a "-snapshot" version to the Jenkins Maven repo.

Version 1.5.2 (29. July 2014)

- Added additional unittests for new features.
- Added REST API URLs that allow the user to get the full configuration overview of a job, including all parents (`http://<jenkins>/job/<name>/getConfigAsXML?depth=1`).
- Fixed rebuild not working for "hidden" parameters, this also adds dynamically created parameters to the rebuild page.
- Save new version of Job, when project is updated via XML upload.
- Made the saving of the versions.xml.gz atomic.
- Fix issue with "hidden" parameters not being hidden from build page, if a "parameter reference" was the last inherited parameter for a given name
- Fixed issue with post-build steps not taking versioning into account
- Added compile-time check for Java6 compatibility
- Fixed issue with altering versioning information not triggering automatic job creation. Now, any save of versioning will trigger creation.
- Fix failure of parameter sanity check, when an inheritable parameter referenced/overwrote a non-inheritable parameter.
- Forcing inheritance, when job properties are evaluated from a BuildStep
- Fixed issue with parameters being derived wrong. When a reference to a reference was selected, the inheritance mode (Overwritable, Extensible, Fixed) was not computed properly.
- Fixed triggers (like the Gerrit or Timed Triggers) not forcing inheritance of values, thus only using locally defined parameters, etc. pp.

Version 1.5.1 (13. April 2014)

- Fixed issue with build flow view not displaying all parents
- Fixed issue with label generation due to caching overriding inheritance. Cached labels now always use fully inherited values.
- Added option to retrieve an XML containing the settings of all parents in one file.
- Reduced usage of the ThreadAssocStore, that should prevent versions from spilling from one request to another, which could cause the `"isBuildable()"` check to fail.
- Restored configuration for SCMs that were missing in v1.5.0
- Added extended build information added to main project page
- Some internal minor code reorganization and bug fixes
- Fixed error that spuriously declared parameter classes as incompatible, thus failing a build before it even started.
- Converted several Jelly to Groovy files and fixed the layout of the build-flow view
- Exchanged names like "Transient Project" with more intuitive ones ("Compounds" in this case)
- The warning messages for fixed parameters now evaluate correctly
- Fixed issue where invalid parameters were not detected, if the final build was not transient.
- Fixed aborting when a variable was declared as "must have a value set".
- Force users to enter a versioning string before saving
- Fixed issue with URL encoding on rebuild-page (and others) -- applies to running behind a URL-rewriting Apache Proxy
- Added a size-limited LRU buffer for decoding the version map. This should slightly speed up accessing specific versions of projects.
- Added notification banner to main project page, to inform the user about the state of versioning of the job (stable/non-stable)
- Fixed Compound configuration screen not using the correct versions of jobs
- Fixed stack-overflow when generating transient actions.

Version 1.5.0 (16. December 2013)

- Great improvements to Java reflection caching. This improves performance of the plug-in quite a bit. (Thanks to Andreas Gudian)

- Fixed issue with the "Keep build forever" button not appearing.
- Added new configuration read/view/download page for project- and build-pages. This allows you to see all build steps and final parameter evaluations on one page; sorted by which project contributed it.
- Added experimental support for inheriting triggers. This feature currently defaults to being disabled; but prints a warning to users on each project page that the new feature is present and can be enabled at any time.
- Fixed issue with several links missing the "context-URL", if Jenkins ran inside a context-ed container.
- SVG Graph now honours order of inheritance. Inverted the direction of the arrows for more clarity of how properties flow from parents to children.
- Added table that shows last builds sorted by build time.
- History table is hidden when a project is not buildable and has no previous runs.
- Fixed issue with non-existing references resetting to default. Now, the old value can be kept around; as the plug-in will detect missing projects anyway.
- Fixed Null Pointer Exception bug in jobs without any parameters.
- Implemented caching of generated labels for Jobs. This leads to a considerable speed boost if all build executors are busy and many jobs queue up. It also fixes a rare issue where labels were not being inherited correctly.

Version 1.4.12 (10. October 2013)

- Hotfix for v1.4.11 -- there was a regression with the "properties not being evaluated in the correct order" bugfix.

Version 1.4.11.1 (09. October 2013)

- Identical to v1.4.11. Re-upload because maven created a snapshot instead of release version

Version 1.4.11 (03. October 2013)

- Fixed issue with properties not being evaluated in the correct order -- some properties were being inherited multiple times, if defined in different parents.
- Layouting fixes and improved support for context-based URLs on many GUI pages
- Replaced many Jelly pages with Groovy scripts
- Changed name of "Child Job Creation" to "Compound Creation".
- Added additional a table to the project overview page to more immediately see direct parents and children that are leafs.
- Moved "project class" from compound-config page to general page. This makes sense, as this field can be used in the "Related Projects View" to sort/filter projects
- Improved rendering of "range" slider element.
- Added option to disallow compound renaming (only relevant for automatically created 'compound' jobs)
- Fixed issue with project-names instead of parameter-names being displayed on "Show Parameter derivation" page

Version 1.4.10 (16. September 2013)

- Added new permission type "Configure Versions". This allows you to split general "Configure" rights and the right to set a version to stable. The "Administer" permission implies the right to configure versions.
- Fixed issue with SubTasks not being inherited. Now, all properties that contributed additional tasks to be processed on other Executors should work correctly.
- Fixed Stack Overflow due to Parameter References. This could occur if you defined the same parameter twice.

Version 1.4.9 (22. August 2013)

- Fixed issue with relative path in Jelly file for "Inheritable Parameter" that caused issue if Jenkins server was a Windows host. (Found and fixed by Steven Craft)
- Removed superfluous scanning of all projects when resolving project references (improves speed on large installations)
- Fixed issue with broken links on th "Show References" page if Jenkins ran in a context other than "/"
- Fixed several possible Null-Pointer-Exceptions in various parts of the plug-in.
- Added several test-cases to prevent future regressions. Do note that these tests are disabled on Windows, due to a bug in Jenkins < 1.520
- Fixed issue with the "magic node-label" (which, if set on a node, makes sure that **only** tests that use that label are scheduled on this machine) being applied wrong if a job had no label set at all.

Version 1.4.8 (01. July 2013)

- Initial public release
- Fixed deadlock occurring in certain situations involving build Queue queries
- Added option to hide "rarely changed" build parameters from the build screen.