

# Promoted Builds Plugin

## Plugin Information

View promoted builds [on the plugin site](#) for more information.



Older versions of this plugin may not be safe to use. Please review the following warnings before using an older version:

- [Unauthorized users are able to run some promotion processes](#)

This plugin allows you to distinguish good builds from bad builds by introducing the notion of 'promotion'. Put simply, a promoted build is a successful build that passed additional criteria (such as more comprehensive tests that are set up as downstream jobs.) The typical situation in which you use promotion is where you have multiple 'test' jobs hooked up as downstream jobs of a 'build' job. You'll then configure the build job so that the build gets promoted when all the test jobs passed successfully. This allows you to keep the build job run fast (so that developers get faster feedback when a build fails), and you can still distinguish builds that are good from builds that compiled but had runtime problems.

Another variation of this usage is to manually promote builds (based on instinct or something else that runs outside Jenkins.) Promoted builds will get a star in the build history view, and it can be then picked up by other teams, deployed to the staging area, etc., as those builds have passed additional quality criteria. In more complicated scenarios, one can set up multiple levels of promotions. This fits nicely in an environment where there are multiple stages of testings (for example, QA testing, acceptance testing, staging, and production.)

## Promotion Action

When a build is promoted, you can have Jenkins perform some actions (such as running a shell script, triggering other jobs, etc. — or in Jenkins lingo, you can run build steps.) This is useful for example to copy the promoted build to somewhere else, deploy it to your QA server. You can also define it as a separate job and then have the promotion action trigger that job.



### Do not rely on files in the workspace

The promotion action uses the workspace of the job as the current directory (and as such the execution of the promotion action is mutually exclusive from any on-going builds of the job.) But by the time promotion runs, this workspace can contain files from builds that are totally unrelated from the build being promoted.

To access the artifacts, use the [Copy Artifact Plugin](#) and choose the permalink.

## Usage

To use this plugin, look for the "Promote builds when..." checkbox, on the Job-configuration page. Define one or a series of promotion processes for the job.

Then, after the promotion processes have been added *and another build is run*, a 'Promotion Status' menu item will be added to the new build's menu options. **Note that this means that builds run before this point cannot be promoted.**

How might you use promoted builds in your environment? Here are a few use cases.

Artifact storage -- you may not want to push an artifact to your main artifact repository on each build. With build promotions, you can push only when an artifact meets certain criteria. For example, you might want to push it only after an integration test is run.

Manual Promotions - You can choose a group of people who can run a promotion manually. This gives a way of having a "sign off" within the build system. For example, a developer might validate a build and approve it for QA testing only when a work product is completed entirely. Then another promotion can be added for the QA hand off to production.

Aggregation of artifacts - If you have a software release that consists of several not directly related artifacts that are in separate jobs, you might want to aggregate all the artifacts of a proven quality to a distribution location. To do this, you can create a new job, adding a "Copy artifacts from another job" (available through Copy Artifact plugin) for each item you want to aggregate. To get a certain promotion, select "Use permalink" in the copy artifact step, then your promoted build should show up in the list of items to copy.

## Notes

### On Downstream Promotion Conditions

One of the possible criteria for promoting a build is "When the following downstream projects build successfully", which basically says if all the specified jobs successfully built (say build BD of job JD), the build in the upstream will be promoted (say build BU of job JU.)

This mechanism crucially relies on a "link" between BD and BU, for BU isn't always the last successful build. We say "BD qualifies BU" if there's this link, and the qualification is established by one of the following means:

1. If BD records fingerprints and one of the fingerprints match some files that are produced by BU (which is determined from the fingerprint records of BU), then BD qualifies BU. Intuitively speaking, this indicates that BD uses artifacts from BU, and thus BD helped verify BU's quality.
2. If BU triggers BD through a build trigger, then BD qualifies BU. This is somewhat weak and potentially incorrect, as there's no machine-readable guarantee that BD actually used anything from BU, but nonetheless this condition is considered as qualification for those who don't configure fingerprints.

Note that in the case #1 above, JU and JD doesn't necessarily have to have any triggering relationship. All it takes is for BD to use some fingerprinted artifacts from BU, and records those fingerprints in BD. It doesn't matter how those artifacts are retrieved either — it could be via [Copy Artifact Plugin](#), it could be through a maven repository, etc. This also means that you can have a single test job (perhaps parameterized), that can promote a large number of different upstream jobs.

Note that after installing this plugin and configuring a promotion process, the option to promote the build will *not* be available for builds run before the promotion process was configured.

## Available Environment Variables

The following environment variables are added for use in scripts, etc. These were retrieved from github [here](#).

- PROMOTED\_URL - URL of the job being promoted
  - ex: [http://jenkins/job/job\\_name\\_being\\_promoted/77/](http://jenkins/job/job_name_being_promoted/77/)
- PROMOTED\_JOB\_NAME - Promoted job name
  - ex: job\_name\_being\_promoted
- PROMOTED\_NUMBER - Build number of the promoted job
  - ex: 77
- PROMOTED\_ID - ID of the build being promoted
  - NOTE: The format changed as a part of [JENKINS-24380](#) in the Jenkins core. Since the original format was used to retrieve timestamps, the new PROMOTED\_TIMESTAMP field has been added in 2.25
- PROMOTED\_TIMESTAMP - timestamp of the promotion. Format example: 2016-02-17T10:46:14Z
  - Date format and Time zone used can be configured in Manage Jenkins -> Configure System Under the Build Pipeline Plugin section

- You might want to review the links below to understand the format you should use:
  - <https://docs.oracle.com/javase/8/docs/api/java/util/TimeZone.html>
  - <https://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html>
- PROMOTED\_USER\_NAME - the user who triggered the promotion
- PROMOTED\_USER\_ID - the user's id who triggered the promotion
- PROMOTED\_JOB\_FULL\_NAME - the full name of the promoted job

## Changelog

Version 3.3 and newer versions

See <https://github.com/jenkinsci/promoted-builds-plugin/releases>

Version 3.2 (JUN 4, 2018)

- [PR #116](#) - Show the full display name of runs in promoted build parameters.

Version 3.1 (Mar 12, 2018)

- [JENKINS-40803](#) - Prevent infinite loop when promoting a build while the Config File Provider Plugin is installed.

Version 3.0 (Feb 26, 2018)

- [SECURITY-746](#) - Make permissions consistent for Approve, Re-Execute, and Force promotion actions
  - Users with just the Promotion/Promote permission are no longer allowed to re-execute or force promotions with a manual condition that specifies a list of users, unless the user is on that list
  - Users specified in a manual promotion condition are now allowed to force this promotion
  - Administrators are now able to approve any promotion with a manual condition

Compatibility Notes:

- This change alters the behavior of the Plugin in some conditions, jobs may require reconfiguration.
- Table below shows the permission changes. Legend:
  - Cells with bold red text - indicate combinations, which revoke dangerous permissions
  - (manual condition) - Action

Jobs which have no *Manual Promotion Condition*

Authorized actions	Granted Permissions			
	Overall/Administer	Promotion/Promote	User/group in the manual condition user list	Job/Read
Approve	N/A	N/A	N/A	<b>N (new)</b>
Re-execute	Y	Y	N/A	<b>N (new)</b>
Force	Y	Y	N/A	N

Jobs which have *Manual Promotion Condition* without users/groups defined:

Authorized actions	Granted Permissions			
	Overall/Administer	Promotion/Promote	User/group in the manual condition user list	Job/Read
Approve	Y	Y	N/A	<b>N (new)</b>
Re-execute	Y	Y	N/A	<b>N (new)</b>
Force	Y	Y	N/A	N

Jobs with a *Manual Promotion Condition* with a non-empty set of users/groups :

Authorized actions	Granted Permissions			
	Overall/Administer	Promotion/Promote	User/group in the manual condition user list	Job/Read
Approve	<b>Y (new)</b>	N	Y	<b>N (new)</b>
Re-execute	Y	<b>N (new)</b>	Y	<b>N (new)</b>
Force	Y	<b>N (new)</b>	<b>Y (new)</b>	N

Version 2.31.1 (Feb 13, 2018)

-  [JENKINS-49433](#) - Prevent NullPointerException in JobDSL when omitting the `evenIfUnstable` argument
  - Affected JobDSL methods: `selfPromotion()`, `parameterizedSelfPromotion()`, `downstream()`
-  [JENKINS-48634](#) - Prevent ClassCastException when running promotion from a status page with empty forms

Version 2.31 (Oct 23, 2017)

-  [JENKINS-47573](#) - Prevent the extension loading issue when the [Rebuild Plugin](#) is not installed (regression in 2.30)

Version 2.30 (Oct 19, 2017)

-  [PR #106](#) - Update Jenkins core requirement to 1.625.3
-  [PR #106](#) - Update Maven plugin requirement to 3.0 and cleanup issues in library dependency conflicts
-  [JENKINS-22068](#) - Fail promotion builds instead of hanging when a promotion is built/rebuilt directly

Version 2.29.1 (Sep 7, 2017)

-  [JENKINS-37368](#) - Improve performance of DownstreamPass promotion condition listener logic

Version 2.29 (Jun 16, 2017)

-  [PR #103](#) - Update the target core baseline to 1.609.3
-  [JENKINS-29586](#) - Add support for ANSI color output in console
- [PR #102](#) - Fix link to the Fingerprint Wiki Page

Version 2.28.1 (Jan 31, 2017)

-  [JENKINS-40552](#) - Prevent possible deadlock between JobPropertyImpl and PromotionProcess

Version 2.28 (Nov 18, 2016)

-  [JENKINS-29358](#) - Fix the sort order of the Promotion History
-  [JENKINS-36623](#) - Fix test runs against the 1.651.3 core (Plugin compatibility tester)
-  [JENKINS-39362](#) - Fix the binary compatibility issues in test runs by getting rid of mockito-all

#### Version 2.27 (May 25th, 2016)

-  [JENKINS-34826](#) - Make the plugin compatible with [SECURITY-170](#) fix in Jenkins 1.651.2+ and 2.3+
-  [PR #91](#) - Prevent exceptions on startup if the optional JobDSL plugin is missing
-  [PR #90](#) - Improve the workspace allocation performance

#### Version 2.26 (May 9th, 2016)

-  [JENKINS-29776](#) - Add Job DSL support for Promoted Builds Plugin
-  [JENKINS-33147](#) - PROMOTED\_TIMESTAMP used to show the UTC timestamp instead of the local JVM one. Behavior is configurable now
-  [PR #76](#) - Allow hiding promotion processes based on a parameter value
-  [PR #89](#) - Migration to the new Jenkins Plugin parent POM
-  [PR #87](#) - Fixes of minor issues discovered by FindBugs
-  [JENKINS-34204](#) - Fix minor issues in Job DSL support discovered during the local snapshot testing (not a regression)

#### Version 2.25 (Feb 19th, 2016)

-  [JENKINS-32993](#) - Expose the PROMOTED\_TIMESTAMP variable in promotion processes
-  [PR #75](#) - Add support for running promotions on inheritance projects
-  [JENKINS-31908](#) - Sort Promotions in UI by Number instead of customization ID
-  [JENKINS-31406](#) - Broken layout of promotion status pages in 1.625+

#### Version 2.24.1 (Dec 17th, 2015)

-  [JENKINS-32006](#) - Revert the erroneous fix of [JENKINS-13751](#) in 2.24

#### Version 2.24 (Nov 25th, 2015)

-  [JENKINS-13751](#) - Promotion process should provide the project's SCM to promotion steps (**reverted in 2.24.1**)
-  [JENKINS-31356](#) - The plugin should consider approvers in the re-execution action
-  [PR #72](#) - Fix exception if somebody calls PromotionProcess::isFingerprintConfigured()
-  [JENKINS-29793](#) - Provide parameters from manual approval on all promotions
-  [JENKINS-27716](#) - Support disabling the promotion jobs in jenkins-1.585+

#### Version 2.23.1 (Nov 2nd, 2015)

-  [JENKINS-31320](#) - Make the plugin compatible with breaking UI changes in Jenkins 1.619+

#### Version 2.23 (Sept 15th, 2015)

-  [JENKINS-30212](#) - Medium-sized promotion star icons were missing
-  [JENKINS-30411](#) - Add the PROMOTION\_ENV token macro

#### Version 2.22 (Aug 25th, 2015)

-  [JENKINS-25011](#) - Fix the Folders support in Promoted builds parameter
  - Now there is a support of global and relative addressing to builds ('.' and '..' markers are supported)
  - See the built-in documentation for more details
  - The change alters the default behavior
-  [JENKINS-7739](#) - Recursively evaluate downstream projects in DownstreamPassCondition
-  [JENKINS-3549](#) - Promotion Status column (last promotions for each type)
-  [JENKINS-3549](#) - Last Build Promotion Status column
-  [JENKINS-16063](#) - Inject PROMOTED\_USER\_ID variable into the build
-  FindBugs cleanup (potential NPEs, concurrency, etc.)



#### Warning!

The fix for [JENKINS-25011](#) changes the default behavior for paths without an explicit absolute/relative address specification (e.g. project="myProject"). By default it will resolve relative paths and then falls back to global paths. If you have "myProject" within a folder and on the top level, another project may be returned after the plugin update. In order to restore the legacy behavior, use the **hudson.plugins.promoted\_builds.util.ItemPathResolver.enableResolutionAgainstRoot** Boolean system property

#### Version 2.21 (Apr 7, 2015)

- [issue #24782](#) Prevent phantom builds from being scheduled when PromotionProcesses are built directly.

#### Version 2.20 (Feb 16, 2015)

- [issue #20492](#) partial fix, show re-execute button unconditionally
- fix an NPE in `getBuilds()` when projectName is incorrect
- added support for rebuild plugin

#### Version 2.19 (Oct 10, 2014)

- Prevent log file being cluttered with permission exceptions when users have `Item.EXTENDED_READ` but not `Item.CONFIGURE`

#### Version 2.18 (Aug 25, 2014)

- [JENKINS-8963](#) CRUD for promotion processes using REST API.
- [JENKINS-23565](#) Export build display name to promotion job.

#### Version 2.17 (Mar 5, 2014)

- Repeating/Duplicating promotion parameters ([issue #22005](#))

#### Version 2.16 (Mar 5, 2014)

- Added `PROMOTED_USER_NAME` environment variable ([issue #16063](#))
- Fixed a couple of typos
- Fixed repeated exception being thrown when installed with `literate` plugin.

#### Version 2.15 (Jan 28, 2014)

- Fixed NPE when no Manual Promotion was configured ([JENKINS-20166](#))

#### Version 2.14

- Enable editing parameters for re-executions ([JENKINS-8962](#))

#### Version 2.13

- Added `PROMOTED_JOB_FULL_NAME` environment variable ([JENKINS-18958](#))

#### Version 2.12 (Aug 20, 2013)

- Expose promotion information via the REST API
- Prevent duplication of promotion processes when creating promotion processes from other plugins

#### Version 2.11 (Jun 09, 2013)

- Fix for an NPE, and diagnosis for another.
- Reduced plugin size by eliminating unnecessarily bundled library.

#### Version 2.10 (Mar 30, 2013)

- NPE when used in 1.507+ ([JENKINS-17341](#))
- Test failures, and possibly visible bugs, when used with recent Jenkins ([JENKINS-15156](#))

#### Version 2.9 (Mar 25, 2013)

- New promoted build parameter that can be used to select builds that have been promoted
- Fixed file parameter on `ManualApproval` not correctly uploading the file

#### Version 2.8 (Oct 30, 2012)

- Build shouldn't be promoted by a deleted promotion process ([issue #12799](#))
- Honor annotated console output ([issue #15328](#))

#### Version 2.7 (Sep 26, 2012)

- Added a trigger that allows projects to listen to promotions happening in other projects
- `PROMOTE` permission can be used in project matrix-based security ([JENKINS-14890](#))
- Downstream jobs textbox is now auto-complete-capable ([JENKINS-14560](#))

#### Version 2.6.2 (Aug 6, 2012)

- Fix manual promotion of maven project and matrix project ([JENKINS-13631](#), [JENKINS-13472](#))
- Fix 404 when clicking the promotion's progress bar to view console output ([pull-20](#))

#### Version 2.6.1 (Jul 2, 2012)

- Fix preventing setting "Restrict where this project can be run" ([JENKINS-14197](#))

#### Version 2.6 (Jun 21, 2012)

- Fixed incorrect link from the executor status (<https://github.com/jenkinsci/promoted-builds-plugin/pull/17>)

- Added a new promotion process that triggers right after a build is completed when it's parameterized. (<https://github.com/jenkinsci/promoted-builds-plugin/pull/16>)
- Fixed a 500 error when viewing promotion on a failed build ([JENKINS-12386](#))
- Image files are now in the PNG format.

#### Version 2.5 (Apr 12, 2012)

- Improved hierarchical project support

#### Version 2.4 (Nov 3, 2011)

- Fixed a possible NPE that fails the promotion ([JENKINS-11609](#))
- Added Promotion History per Promotion Process at Project's Promotion Status page ([JENKINS-10448](#))

#### Version 2.3.1 (Oct 14, 2011)

- Don't run promotionProcess that are disabled ([JENKINS-10423](#))
- Manual approvement causes an NPE if project name or promotion name contains URI-unsafe chars ([JENKINS-11122](#))

#### Version 2.3 (Aug 9, 2011)

- Modified the self-promotion condition so that it does not trigger for builds which are a failure. Also it is now configurable whether to self-promote for unstable builds. ([JENKINS-10250](#))

#### Version 2.2 (Jul 8, 2011)

- Added a new promotion condition that immediately promotes itself.

#### Version 2.1 (May 4, 2011)

- failed to Re-execute promotion if promotion builds plugin is disabled ([JENKINS-9588](#)).
- promote plugin should provide ability to select slave node to run ([JENKINS-9260](#)).
- Fix for NPE when promoting a build with a custom workspace ([JENKINS-9254](#)).
- Fixed a problem where removing a promotion process leaves broken image links in the build history.
- Fixed a problem where deleting and recreating the promotion process with different case (abc vs ABC) can result in weird behavior on Windows.

#### Version 2.0 (Mar 5 2011)

- If a promotion criteria is met but the promotion fails, change the icon to represent that.
- Exposed the job name and the build number of the promotion target to the promotion process (PROMOTED\_JOB\_NAME and PROMOTED\_NUMBER.)
- If the build is parameterized, expose that to the promotion process as well
- Added a new promotion criteria where a promotion in upstream promotes a downstream build.
- Fully implemented manual approval with user / group permissions and display Approve button on promotion page (no longer need to allow force promotion to all)
- New promotion action to mark the promoted build as keep forever

#### Version 1.11 (Jan 31 2011)

- Promote a build even if downstream build is unstable. ([JENKINS-8626](#))
- Promote Builds Plugin can use custom workspace. ([JENKINS-8547](#))
- Invalid characters in Promotion name causes error. ([JENKINS-7972](#))
- Fix promotion permlinks. ([JENKINS-8367](#))
- Allow promotion actions to be reordered. ([JENKINS-8548](#))

#### Version 1.10 (Sep 28 2010)

- Promotion processes are now recognized as permalinks.

#### Version 1.9 (Jun 9 2010)

- If fingerprints are not available, use the upstream/downstream triggering information to determine the relationship as a fallback.

#### Version 1.8 (Jun 5 2010)

- Add the possibility to choose a different color for the star icon, to be able to differentiate the various promotion processes

#### Version 1.7 (Mar 29 2010)

- Use JDK configured for project when running promotions ([JENKINS-3526](#))
- Select node for running promotions from label configured for project ([JENKINS-4089](#)) (does not yet run on exact node where promoted build ran, unless project is tied to a single node)
- Show most-recent first in promotion history tables ([JENKINS-6073](#))

#### Version 1.6 (Dec 30 2009)

- Fix for running on slave node ([JENKINS-4635](#))
- Copy promotions when a job is copied ([JENKINS-3489](#))
- Fix broken "Back to Promotion Status" link ([JENKINS-3562](#))

#### Version 1.5 (Aug 17 2009)

- Updated to work with Jenkins 1.320

#### Version 1.4 (May 21 2009)

- Re-doing a release as 1.3 had never shown up in the update center.

#### Version 1.3 (May 11 2009)

- Expose environment variable 'PROMOTED\_URL' that points to the URL of the build that's being promoted ([report](#))
- Internal modernization.

#### Version 1.2 (Mar 25 2009)

- Updated to work with the current Jenkins

#### Version 1.1 (Feb 20 2009)

- Fixed a problem where Jenkins may issue the same warning multiple times for the same configuration problem ([report](#))
- SVN Tagging is now a permitted promotion step
- Promotion now fails if any actions are not performed ([JENKINS-2597](#))
- Improved logging of promotion build process so users can see what succeeded and what failed
- Promotion action to build another project no longer does nothing ([JENKINS-1765](#))
- Added "Promotion History" pane to the PromotedBuildAction page ([JENKINS-2794](#))
- Fixed 404 for last failed link while promotion build occurring ([JENKINS-2578](#))`PROMOTED_URL`