EnvInject Plugin

Plugin Information
View Environment Injector on the plugin site for more information.

The current version of this plugin may not be safe to use. Please review the following warnings before use:

- Exposure of sensitive build variables stored by EnvInject 1.90 and earlier

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

- Low privilege users are able to read parts of some files on master
- Arbitrary code execution vulnerability

This plugin is up for adoption. Want to help improve this plugin? Click here to learn more!

This plugin makes it possible to have an isolated environment for your jobs.

Summary

EnvInject plugin provides the following features:

- Removes inherited environment variables by the Jenkins Java process
- Injects environment variables at node (master/slave) startup
- Executes a setup script before or/and after a SCM checkout for a run
- Injects environment variables before or/and after a SCM checkout for a run
- Injects environment variables as a build step for a run
- Injects password values for a run
- Exports environment variables at the end of the build in order to to know the set of environment variables used for each build

Warning!

This plugin has some known limitations. For Example, Pipeline Plugin is not fully supported. See the documentation below for more details.

Table of contents

Some use cases

- To remove inherited environment variables (PATH, ANT_HOME, ...) at node level (master/slave), available by default for a job run.
- To inject variables in the first step of the job (before the SCM checkout)
- To inject variables based on user parameter values
- To execute an initialization script before a SCM checkout.
- To execute an initialization script after a SCM checkout
- To inject variables as a build step obtained from a file filled in by a previous build step
- To know environment variables used for a build
- To inject build cause as environment variable for a build
- To inject environment variables from the evaluation of a Groovy script (powered by Script Security Plugin)
- To export environment variables as a metadata in your binary repository

Build Causes

This plugin also exposes the cause of the current build as an environment variable. A build can be triggered by multiple causes at the same time e.g. an SCM Change could have occurred at the same time as a user triggers the build manually.

The build cause is exposed as a comma separated list:

```
BUILD_CAUSE=USERIDCAUSE, SCMTRIGGER, UPSTREAMTRIGGER, MANUALTRIGGER
```

In addition, each cause is exposed as a single envvariable too:

```
BUILD_CAUSE_USERIDCAUSE=true
BUILD_CAUSE_SCMTRIGGER=true
BUILD_CAUSE_UPSTREAMTRIGGER=true
BUILD_CAUSE_MANUALTRIGGER=true
```

Configuration

At node (master and agent) level

At job level

Set up a clean environment

☑ Prepare an environment for the job ⓘ

Keep Jenkins Environment Variables ☐ ⓘ

Keep Jenkins Build Variables ☑ ⓘ

Script File Path [   ] ⓘ

Script Content
```
[                                                    ]
```
ⓘ

Properties File Path [/var/public/jenkins/envVars.properties] ⓘ

Properties Content
```
[                                                    ]
```
ⓘ

Load files from the master ☐ ⓘ

Inject variables after a SCM checkout

Build Environment

☑ Inject environment variables to the build process ⓘ

Properties File Path [   ] ⓘ

Properties Content
```
ANT_HOME=/usr/ant
```
ⓘ

Script File Path [   ] ⓘ

Script Content
```
[                                                    ]
```

Inject variables as a build step

Build

⠿ **Inject environment variables for your job** ⓘ

Environment Properties File Path [   ] ⓘ

Environment Properties Content
```
DEPLOYED_DIR=/user/tc/deployedWebapps
```
ⓘ

[Delete]

[Add build step ▾]

Execute Windows batch command
Execute shell
Invoke Ant
Invoke top-level Maven targets
Inject environment variables for your job
Always fail
Capture Environment Variables
Force the build result
Sleep
Make build unstable

☐ Record fingerprints of files to track usage ⓘ

☐ E-mail Notification ⓘ

Variables Traceability

Each build captures environment variables and stores them in an environment file called 'injectedEnvVars.txt' located in $JENKINS_HOME/jobs/<your_job>/builds/<your_build>.

For an old build, you can display injected environment variables through the right UI menu on the build page.

You can also get build environment variables by the following HTTP GET URL: <jenkins_url>/job/<job_name>/<build_number>/injectedEnvVars/export
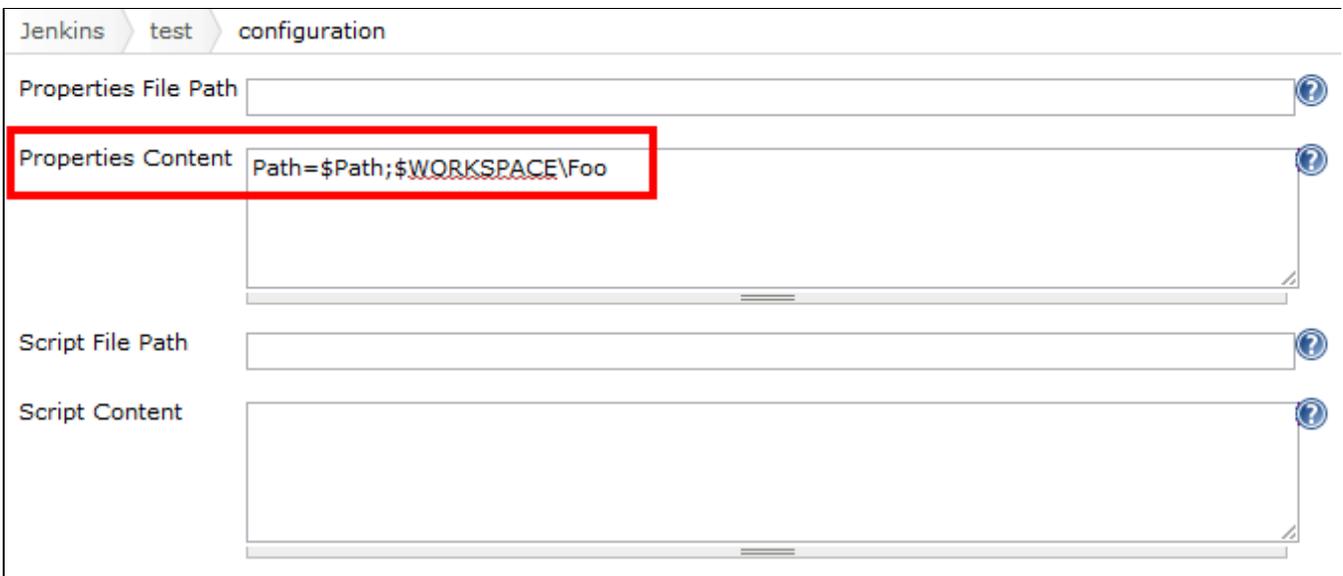XML, JSON and TXT format are supported.
For example:

- curl -X GET -H"Accept:text/plain" <jenkins_url>/job/<job_name>/lastBuild/injectedEnvVars/export
- curl -X GET -H"Accept:application/xml" <jenkins_url>/job/<job_name>/18/injectedEnvVars/export
- curl -X GET -H"Accept:application/json" <jenkins_url>/job/<job_name>/18/injectedEnvVars/export

Examples

Append to PATH on Windows

Warning!

This screenshot is from an older version of EnvInject. Starting with EnvInject 1.93, any backslashes (such as Windows directory separators) in environment variables should be escaped (doubled). (See JENKINS-31573.)

Gerrit Trigger plugin with parameterized Build branch name

- Check "**This build is parameterized**"
    - Name: GERRIT_BRANCH
    - Default Value: master
- Check "**Prepare an environment for the run**"
    - **Evaluated Groovy script**:

```
if (!binding.variables.containsKey('GERRIT_REFSPEC')) {
    return [GERRIT_REFSPEC: "refs/heads/$GERRIT_BRANCH"]
}
```

Additional features provided by other plugins

Shared Objects plugin contributes to Envinject plugin that enables you to populate shared objects as environment variables.

Extensibility with other plugins

EnvInject captures build environment variables populated by plugins providing environment variables through Jenkins extension points (such as BuildWrappers, EnvironmentContributions and so on).
Environment variables injected by the EnvInject plugin are available in Jenkins triggers (for example in all XTrigger plugin typologies, injected environment variable can be used).
Injected environment variables with the EnvInject plugin are captured by the BuildContext capture plugin.

Comparison with other plugins

This plugin is an alternative to Setenv Plugin and Envfile Plugin plugins
Note 1: The EnvInject plugin automatically migrates the Jobs configuration from these plugins. The setenv and/or the envfile plugins can be deleted from your plugin list.
Note 2: Master and slave are managed.

Known incompatibilities

Jenkins Pipeline

Even though it is possible to set up the EnvInject Job Property and build step in Pipeline, the plugin does not provide full compatibility with Pipeline Plugin.

Supported use-cases:

- Injection of EnvVars defined in the "Properties Content" field of the Job Property
    - These EnvVars are being injected to the script environment and will be inaccessible via the "env" Pipeline global variable
    - Please note there is also a known compatibility issue with Durable Task Plugin 1.13

All other use-cases of the plugin may work incorrectly in Jenkins Pipeline. Please see JENKINS-42614 for more information about unsupported use-cases. There is no short-term plan to fix the compatibility issues though any pull requests to the plugin will be appreciated.

Mask Passwords Plugin

MaskPasswords plugin is able to mask password elements. However, technically, this plugin provides its own class.

And we don't want to have a hard dependency from the EnvInject plugin to any other plugin. Therefore, EnvInject can't retrieve information from the MaskPassword plugin. However, EnvInject provides all MaskPasswords features.

Tool Environment Plugin

EnvInject can't use exported environment variables provided by the Tool Environment plugin.
Instead, we suggest to use the SharedObjects Plugin. It covers Tool Env plugin features and provides a good integration with EnvInject.

Known limitations

- As into regular shell scripts, you are not able to use "." character for environment variable names

- Only previous environment variables are available for polling
  Some plugins provide polling mechanisms (such as SCM plugins, XTrigger plugins, ...) and you want to use injected environment variables in form fields of these plugins. Unfortunately, injected environment variables are processed only at build time therefore after the build is scheduled. Therefore, we can't access environment variables configured within the job. However, previous injected environment variables (from the previous build) are retrievable (implemented for example in the XTtrigger plugins). For the other plugins, authors are free to add the envinject-lib library dependency in their plugins in order to provide the ability to use environment variables given the EnvInject plugin.

Changelog

2.1.6 (Jul 07, 2018)

Release with minor improvements in the codebase:

- PR #131 - refactor the code to prevent warnings about attempts to (de-)serialize anonymous classes in Jenkins 2.107.2+
- PR #129 - update the plugin so that it can be tested with Plugin Compat Tester
- PR #130 - fix typo in the Groovy script variable documentation

2.1.5 (Oct 13, 2017)

- Add compatibility notice for issues reported in 2.1.4

2.1.4 (Oct 07, 2017)

> **Warning!**
>
> There are two regression issues reported: JENKINS-47364, JENKINS-47370
>
> - Variable substitution stopped work in some cases
> - It is a regression after the JENKINS-26583 patch
> - In particular, it affects Jenkins Pipeline which is not covered by autotests
>
> The fix will be published soon. The plan is to make the newly introduced behavior opt-in

- JENKINS-26583 - The plugin does not longer override Environment Variables from plugins by the values persisted in its cache.
  - This update does not solve all potential conflicts between EnvInject and other environment-contributing plugins
- JENKINS-46479 - Prevent environment variables from being deleted on existing builds when the action is saved before cache initialization.
- Update EnvInject Lib from 1.26 to 1.27 in order to pick FindBugs fixes and JENKINS-46479.

2.1.3 (July 3, 2017)

- JENKINS-32428 - Fix glitches in JSON escaping in REST API (regression in 2.1.1)

2.1.2 (Jun 28, 2017)

- JENKINS-45056 - Fix issue with the incorrect EnvInject API Plugin 1.0 packaging. Now there is a plugin dependency
- JENKINS-45055 - Update EnvInject Lib from 1.25 to 1.26 in order to pick FindBugs fixes (changelog)

2.1.1 (Jun 20, 2017)

- JENKINS-32428 - Escape XML and JSON outputs in REST API for "Injected Environment Variables"
- JENKINS-44263 - Prevent showing empty "Injected Environment Variables" list after build loading from the disk due (reliance on the cached value)

2.1 (May 12, 2017)

This is an API release, which is a preparation for a better Jenkins Pipeline support in the plugin (JENKINS-42614).

- JENKINS-43845 - Create new EnvInject API plugin, which provides new Pipeline-compatible API of EnvInject Lib and EnvInject API util classes
- JENKINS-43536 - Rework internal logic to avoid using AbstractProject explicitly where possible
- JENKINS-43535 - Deprecate old API in EnvInject Lib and provide migration guidelines

Expected behavior changes:

- AbstractProject types (Freestyle, Matrix, JobDSL, etc.) - nothing changes in the behavior
- Jenkins Pipeline - some use-cases may start or stop working. Jenkins Pipeline stays untested (see Known Limitations)

2.0 (Apr 10th, 2017)

This is a security release of the plugin.

- SECURITY-256 - Arbitrary Groovy code execution vulnerability
- SECURITY-348 - Low privilege users are able to read parts of some files on master

> **Warning!**
>
> This plugin release is not fully compatible with Previous EnvInject versions. A manual job configuration update may be required. See the referenced issues for more information.

1.93.1 (Sep 30th, 2016)

- JENKINS-31829 - Add currentListener Groovy variable
- JENKINS-38607 - Jenkins Update Center now shows compatibility warnings when upgrading from versions below 1.93 (See the compatibility notice regarding JENKINS-31573)
- PR #109 - Incorrect initialization of the field in EnvInject build wrapper with Deprecated API. No user-visible impact
- PR #109 - Escape all plugin Jelly views by default
- PR #107 - Upgrade Jenkins Test Harness framework to the latest version, improve test coverage

1.93 (Sep 26th, 2016)

- Update to the new parent POM, 1.609.3 is a new target core version
- JENKINS-31573 - Incorrect parsing of newline characters
- JENKINS-30028 - Timestamper was displaying shifted timestamps due to the integration issue with EnvInject
- JENKINS-36545 - WORKSPACE variable was unavailable in Groovy script injector
- JENKINS-36466 - Update env-inject-lib to 1.24 in order to avoid potential incompatibilities on core versions with detached and deleted Matrix Project plugin
- JENKINS-32693 - Fix BuildCauseRetriever when a build has null cause action
- Cleanup issues reported by FindBugs: tolerance against node disconnection, etc.
- JENKINS-23274 - Do not output contents of the "Evaluated Groovy script" to console
- Document the current behavior of Node and Jenkins properties (JENKINS-23666)
- Improve diagnostics of issues during the environment setup in EnvInjectBuildWrapper (e.g. JENKINS-36237)

Compatibility notes:

- Fix of JENKINS-31573
  - The change introduces a new property file parsing engine, which starts following the standard for backslashes
  - You may need to update the property files (e.g. convert backslash entries to double backslash)

1.92.1 (Aug 21th, 2015)

- JENKINS-27382 - EnvInjectPluginAction::buildEnvVars() injects masks instead of passwords to the environment (regression since 1.90)
  - There are many similar issues in JIRA, which have been probably fixed by the patch

1.92 (Aug 18th, 2015)

- JENKINS-29817 - Contribute Job Property Contents to the project environment (polling, etc.)
- JENKINS-29867 - Permissions engine + global option for disabling the Injected vars
- Fix several issues discovered by FindBugs

1.91.4 (Aug 4th, 2015)

- Fix JENKINS-27665 and JENKINS-27363 - Prevent NPEs when handling null lists of sensitive variables (passwords, etc.)
- Fix JENKINS-29667 - Text description in the plugin manager was bold

1.91.3 (May 15th, 2015)

- Fix JENKINS-28409 - File descriptor leak in "Inject Passwords" build wrapper
- Fix JENKINS-28188 - StringIndexOutOfBoundsException when no root cause for the build

1.91.2 (Apr 14th, 2015)

- Fix JENKINS-27496 - NPE during submission of EnvInject JobProperty when the form is empty (regression, since 1.91)

1.91.1 (Mar 11th, 2015)

- Fix JENKINS-27342 - NPE in EnvInjectPluginAction.transformEntry() when there is no sensible variables (regression, since 1.91)

1.91 (Mar 8th, 2015)

* Fix JENKINS-19852 - NPE during submission of EnvInject JobProperty configurations w/o admin permissions
* Fix JENKINS-19222 - EnvInject undefines NODE_NAME environment variable on master
* Fix JENKINS-23447 - Mask sensitive data in injectedEnvVars.txt when displayed on UI or persisted on disk
* Fix JENKINS-24785 - BUILD_CAUSE is always UPSTREAMTRIGGER in multi-configuration jobs

> **Warning!**
>
> There are showstopper bugs reported to this version. Please avoid the upgrade till the next release

1.90

* JENKINS-24130 - Added an option to mask PasswordParameters
* Execute Groovy scripts to inject environment variables in Build Wrappers
* Fix JENKINS-22126 - Override build parameters only when explicitly requested
* EnvInject escapes a sensitive data from other Environment contributors
* Fix JENKINS-22169 - Migration from Setenv Plugin overwrites existing envinject configurations

1.89

* Don't mask stacktraces on errors
* Upgrade to envinject-lib-1.19

1.88

* Fix JENKINS-16316 - Changes to global variables not honored
* Log both source and resolved unreachable properties file paths
* Fix JENKINS-14144 - use a WorkspaceListener to avoid injecting a temporary

1.87

* Actually load script from master

1.86

* Fix reopen JENKINS-13348 - EnvInject overriding WORKSPACE variable

1.85

* Retain line break characters in property values.

1.84

* Add logger to Groovy script evaluation

1.83

* Fix reopen JENKINS-16316 - Global variables not updated

1.82

* Fix JENKINS-16566 - poll SCM not work after upgrade the EnvInject Plugin to 1.81
* Fix JENKINS-16575 - SEVERE: Failed to record SCM polling - java.lang.NullPointerException - EnvInject Plugin
* Update to envinject-lib 1.15

1.81

* Fix JENKINS-16239 - NPE in getRootDir
** Update to envinject-lib 1.14
* Remove Injected from Labels
* Fix JENKINS-16380 - Starting with >1.73 global environment variables overrule build specific parameter variables
* Fix JENKINS-16372 -Password parameter is malformed

1.80

* Fix JENKINS-16233 - EnvInject plugin using a cached value for ${WORKSPACE}

1.79

* Fix JENKINS-16316 - Global variables not updated
* Fix JENKINS-16399 - SCM documentation is incorrect

1.78

* Fix JENKINS-16219 regression from JENKINS-14437 - Version 1.77 injects all **environment** variables into ANT command line (-D)

1.77

* Fix JENKINS-14437 - envinject fails to "really" set/override build parameters

1.76

* Fix JENKINS-16016 - Global passwords are visible for matrix configuration builds

1.75

* Add currentJob and currentBuild variables for Groovy scripts

1.74

* Correct spelling of Password from Password (merged from pull request)

1.73

* Fix JENKINS-15658 - EnvInject Undefines Jenkins Global properties referencing WORKSPACE variable
* Fix JENKINS-15664 - Help text for EnvInject plugin is wrong

1.72

* Fix JENKINS-15146 - EnvInject unsets empty string properties returned in maps

1.71

* Support for new job types
* Replace "white-list" strategy for supported job types by use of BuildableItemWithBuildWrappers interface to access the job's buildWrapper list.

1.70

* Fix JENKINS-15071 - Editing the description of Jobs throws an Error 500 message
* Update to envinject-lib 1.13

1.69

* Fix JENKINS-14930 - Can't overload/update PYTHONPATH

1.68

* Fix JENKINS-14897 - ConfigFileProvider variable is not seen by EnvInject
* Require Jenkins 1.444+

1.67

* Fix JENKINS-14761 - Backslash broken on Properties Content injection
* Fix JENKINS-14768 - Backslash is escaped from injected $WORKSPACE property on slave node

1.66

* Fix JENKINS-14686 - Fail evaluating Groovy script before job run

1.65

* Add feature - Mask job password parameter value

1.64

* Fix JENKINS-14645 - Executing external job results in an ClassCastException (ExternalRun cannot be cast to AbstractBuild)

1.63

* Fix reponed JENKINS-14371 - NullPointerException in EnvInjectBuilderContributionAction.buildEnvVars during SCM poll
* Fix JENKINS-14459 - Failed to record SCM polling with envinject plugin

1.62

* Fix JENKINS-14371 - NullPointerException in EnvInjectBuilderContributionAction.buildEnvVars during SCM poll

1.61

* Fix JENKINS-14367 - Global & node level self-referencing variables should work

1.60

* Fix JENKINS-14290 - Support for Ivy project type

1.59

* Fix JENKINS-14271 - EnvInject claims global env vars are unresolved on slave

1.58

* Fix JENKINS-14284 - Single backslashes are droped or used as escape char

1.57

* Fix JENKINS-14232 - EnvInject not handling properties file with logical lines

1.56

* Load environment variables from the node (therefore, you are able to redefine classic variables such as the PATH variable at node level)
* Upgrade to envinject-lib 1.8

1.55

* Technical release: Merge pull request - Make EnvInjectPluginAction implements EnvironmentContributingAction

1.54

* A few minor changes to keep jdk5 compatibility

1.53

* Fix regression on backward compatibility

1.52

* Remove env variables presence in build serialization file

1.51

* Add a complement to fix JENKINS-12423 - Password masked by Mask Passwords are visible when using envinject plugin (no display environment variables when the job is running)

1.50

* Update to envinject-lib 1.7 (Enable to external plugin to retrieve envVars)

1.49

* Merge pull request - use UnsupportedOperationException instead of UnsupportedMediaException
* Merge pull request - added Japanese localization

1.48

* Fix JENKINS-12878 - manage old data fails with RuntimeException: Failed to serialize hudson.model.Actionable#actions for class hudson.model.FreeStyleBuild

1.47

* Fix JENKINS-13566 - EnvInject is messing around with my TEMP variable when username contains a dollar sign (windows)
* Update to envinject-lib 1.5

1.46

* Fix interaction with multijob plugin

1.45

* Add /export/txt, /export/xml and /export/json for the exportation

1.44

* Fix EnvInject Password process

1.43

* Fix reopened JENKINS-12108 - EnvInject failure using multi-configuration jobs

1.42

* Fix JENKINS-13183 - SCM variables resolution

1.41

* Fix JENKINS-13157 - EnvInject cannot inject variables whose contents contain a dollar sign ($)

1.40

* Fix JENKINS-13167 - Cannot use EnvInject in a Maven 2 Jenkins project

1.39

* Fix JENKINS-13155 - Parameters are not working in EnvInject plugin 1.38

1.38

* Fix JENKINS-13119 - Add feature: Set an environment variable based on value of user passed parameter

1.37

* Fix JENKINS-13085 - Environment Variable Injection doesn't work when project run on slave node that sets the same variable

1.36

* Fix JENKINS-13041 - PATH variable is being injected from master on multiconfiguration jobs

1.35

* Fix JENKINS-13022 - EnvInject 1.33 doesn''t seem to work with Jenkins 1.454

1.34

* Change Environment contributors resolution

1.33

* Move global passwords injection to the 'Prepare job environment' section
* Add 'Inject job passwords'

1.32

* Fix JENKINS-12944 - Env Inject Plugin doesn't substitute ${WORKSPACE} variable at all when used in 'Preparing an environment for the job'
* Fix JENKINS-12963 - EnvInject plugin causes job to use JAVA_HOME instead of configured JDK

1.31

* Fix JENKINS-12936 - EnvInject Plugin 1.30 does not display all properties on job config page

1.30

* Fix JENKINS-12905 - PATH variable is not injected

1.29

* Fix JENKINS-12876 - Version 1.20 and later removed all comments and new-lines
* Add test to JENKINS-12841 - EnvInject Plugin 1.26 does not substitute ${WORKSPACE} correctly

1.28

* Fix reponed JENKINS-12108 - EnvInject failure using multi-configuration jobs
* Fix JENKINS-12841 - EnvInject Plugin 1.26 does not substitute ${WORKSPACE} correctly

1.27

* Add a build context to EnvInjectContributor

1.26

* Fix JENKINS-12704 - WORKSPACE variable for concurrent builds are not defined properly.

1.25

* Fix JENKINS-12809 - Injecting from file removes backslashes on ${WORKSPACE}

1.24

* Add the ability to inject password values
* Fix JENKINS-12423 - Password masked by Mask Passwords are visible when using envinject plugin

1.23

* Fix JENKINS-12788 - Missing variables when executing EnvInject script

1.22

* Fix NullPointerException for a setenv or a envfile migration

1.21

* Fix JENKINS-12746 - envinject strips off "\" each time config page is opened

1.20

* Move properties content serialization to a Map (avoid carriage return issues)
* Fix usage of build variables for the 'Prepare environment jobs'
* Fix Injected environment vars when EnvInject is not configured

1.19

* Fix reopened JENKINS-12423 - Password masked by Mask Passwords are visible when using envinject plugin

1.18

* Fix JENKINS-12691 - Property variable not set if using a property file

1.17

* Fix migration from the setenv and envfile plugin

1.16

* Fix external EnvInjectContribution section display
* Fix text typo

1.15 (technical release)

* Refactoring + fix wrong commit

1.14

* Fix JENKINS-12416 - Environment is not injected for each configuration in a multi-configuration project

1.13

* Add the extension point EnvInject Job Property Contributor to enable other plugins to contribute to this plugin for the prepare job step

1.12

* Fix JENKINS-12423 - Password masked by Mask Passwords are visible when using envinject plugin

1.11

* Update to envinject-lib 1.2

1.10

* Update to envinject-lib 1.1
* Fix bug on the build end

1.9

* Export reuse code to a dedicated library: envinject-lib 1.0
* Fix typo

1.8

* Switch properties section and scripts section for the prepare step

1.7

* Fix JENKINS-12293 - Incorrect inserting environment variables on linux slave

1.6

* Recording EnvVars contributing by other plugins at the end of the build

1.5

* Fix JENKINS-12252 - envinject does not fail the build after Environment Script Content fails

1.4

* Fix portability in the history build.xml file (using the build reference instead of the absolute path of the EnvInjected file)
* Add the ability to load a properties file at node startup (at master and/or at slave startup)

1.3

* Move 'Triggered Build Cause' to Jenkins build variables option
* Add an export option to export environment variables to TXT, XML or JSON

1.2

* Fix NullPointerException when master has 0 executors

1.1

* Fix JENKINS-12108 - EnvInject failure using multi-configuration jobs
* Fix JENKINS-12084 -Variables resolved alphabetically rather than order found in content/file. ${WORKSPACE} not always getting resolved as result
* Accept script parameters to script path sections

1.0

* Fix JENKINS-12027 - ${WORKSPACE} variable doesn't use my specific workspace directory anymore in a build step
* Make it stable for production

0.26

* Enhance display log
* Refactoring for fixing properties variables propagation

0.25

* Fix JENKINS-11763 - Can not set some 'special' variable names the first time
* Technical Refactoring

0.24

* Make it extensible by an another plugin.

0.23

* Fixed JENKINS-11595 - Some env variables can not be used to inject, for example $NODE_NAME

0.22

* Add the ability for the 'prepare environment' feature to load the properties file and the script file from the master node even if the build runs on a slave node.

0.21

* Add global properties and node properties process

0.20

* Fix JENKINS-11439 - Can't add "environment inject" to pre-scm-buildstep plugin.

0.19

* Fix JENKINS-11181 -Variables failing to be set if referencing previous variable

0.18

* Fix JENKINS-11067 - Pre-SCM and Post-SCM environment scripts do not use Pre-SCM and Post-SCM properties

0.17

* Fix JENKINS-11066 - Property Variables "Randomly" Not Working/Getting Unset

0.16

* Fix JENKINS-11063 - Multiple "Inject Environment Variables" Build Steps All Show Same Content

0.15

* Backslashes in file path for windows platform are managed
* Added feature 'unset variable' when the variable is unresolved

0.14

* Integrate domi pull request (add trigger causes as environment variables)
* Fix JENKINS-10980 - EnvInject to optinally trim trailing spaces

0.13.1

* Improve reporting errors

0.13

* Fix JENKINS-10919 - Variables are shared between jobs

0.12

* Fixed JENKINS-10916 - Properties not injected and build marked as failed

0.11.1 (technical release)

* Remove unused maven dependency

0.11

* Fix JENKINS-10894 - Inject environment variables with property file not working

0.10

* Add the ability to unset system variable at node level
* Add an option to 'Keep Jenkins system variables'
* Fix JENKINS-10845 and JENKINS-10877 - master/slave sync / Job variables/parameters are missing
* Complete the fix to JENKINS-10847

0.9

* Fix partially JENKINS-10847 and JENKINS-10845 - Enivronment is not separated
* The ability to 'Keep system variables' has been removed. The feature will be restored later (and moved at slave/node level (not at job level).

0.8

* Fixed variables restauration for Maven jobs

0.7

* Fixed variables propagation between typologies
* Fixed variables restore for all typologies

0.6

* Add a 'Keep Jenkins Build variables' option
* Add 'EnvInject' as prefix for plugin log messages
* Sort the EnvInject variables table

0.5

* Remove advanced sections

0.4

* Add a build wrapper
* Fix serialization fiels for JobProperty
* Add a migration procedure for setenv and envfile plugins

0.3

* Add variables resolution in elements

0.2.1

* Fix environment script content process

0.2

* Can inject environment variables as a build step.

0.1

* Initial versio