

The Continuous Integration Game plugin

Plugin Information

View Continuous Integration game [on the plugin site](#) for more information.

This plugin introduces a game where users get points on improving the builds. The plugin is a simple proof of concept of the Build game that was created by [Clint Shank](#). The idea behind the game is to decrease the number of times a build becomes broken. To win the game the developers have to commit code that is compiling and keep adding unit tests.

Usage

Configuration



Version requirements

Note that the plugin version 1.13+ requires Hudson 1.320 to work and the following plugin versions:

- Checkstyle 3.1
- Findbugs 4.0
- PMD 3.1
- Tasks 4.0
- Warnings 3.0
- Violations 0.5.4
- analysis-core 1.0

The plugin is not activated for all jobs at start, each separate job has to activate the game. The game can also be de-activated in one job if some large merge activity is going to take place. To activate the game for a job, go to the job configuration page, click "Add post-build action" and select "*Continuous integration game*" from the list of available post-build actions.

Build score card

For each build the game calculates, according to the rules, the number of points that it is worth. Each user that is responsible for the build (ie comitters) receives the points.



The build was worth 1.0 points.

Clicking on the link shows what rules that were involved in the point calculation.

Score card

Participating players in the build was bob

Rule ↓	Points
Basic ruleset - Build result	-10.0

Leader board

To see who is winning the build game, there is a leader board that is shown on the front page.

Game leader board

Participant	Score ↑
spirou	45.0
bob	25.0
smith	3.0
dave	-52.0

Rules

The rules of the game are:

- -10 points for breaking a build
- 0 points for breaking a build that already was broken
- +1 points for doing a build with no failures (unstable builds gives no points)
- -1 points for each new test failures
- +1 points for each new test that passes

Rules that depend on other plugins:

- [PMD Plugin](#). Adding/removing a HIGH priority warning = -5/+5. Adding/removing a MEDIUM priority warning = -3/+3. Adding/removing a LOW priority warning = -1/+1.
- [Task Scanner Plugin](#). Adding/removing a HIGH priority task = -5/+5. Adding/removing a MEDIUM priority task = -3/+3. Adding/removing a LOW priority task = -1/+1
- [Violations Plugin](#). Adding/removing a violation = -1/+1. Adding/removing a duplication violation = -5/+5.
- [Findbugs Plugin](#). Adding/removing a HIGH priority findbugs warning = -5/+5. Adding/removing a MEDIUM priority findbugs warning = -3/+3. Adding/removing a LOW priority findbugs warning = -1/+1
- [Warnings Plugin](#). Adding/removing a compiler warning = -1/+1.
- [Checkstyle Plugin](#). Adding/removing a checkstyle warning = -1/+1.

Adding rules to the game

Currently there are three ways to add rules to the game.

Patch

Send me a patch for the rule implementation, and I can include it in the main game plugin.

Including rules in another plugin

You are a maintainer of a plugin and would like to add rules to the game with data from your plugin. To do this you should declare the game plugin as an [optional dependency](#) to your plugin. To create rules implement the interface **Rule**, and group them together in a **RuleSet**. To add a **RuleSet** to the game, add it to the game's **RuleBook**.

```
hudson.plugins.cigame.PluginImpl.GAME_PUBLISHER_DESCRIPTOR.getRuleBook().addRuleSet(pluginruleset);
```

If there are already rules for your plugin in the game plugin, let me know so they can be removed from the plugin.

Include rules in a third plugin

You would like to add rules to the game that are only valid for your organization and do not want to share them. To do this you should declare the game plugin as a [mandatory dependency](#) to your plugin. Then implement the **Rule** and add them to the **RuleBook** as in above point.

Changelog

Version 1.26 (2016-10-21)compatibility with DSLs (pull-19)

Version 1.26 (2016-10-21)

- compatibility with DSLs ([pull-19](#))

Version 1.25 (2016-06-30)

- Support for jacoco coverage ([pull-16](#))

Version 1.24 (2016-03-12)

- Support for maven findbugs scores ([pull-18](#))

Version 1.23 (2015-11-08)

- Prevent user to edit score value ([pull-17](#))

Version 1.22 (2015-08-21)

- Expose leaderboard via API ([pull-9](#))

Version 1.21 (2015-02-17)

- Adapt to API changes introduced in Jenkins core, bump version dependencies ([issue #24478](#))

Version 1.20 (2014-07-19)

- Remember history per user where most recent awarded points are coming from ([pull-7](#))
- Correct help link for case insensitive setting ([issue #20125](#))
- Configurable points to be awarded for matching rules ([issue #6446](#))
- leaderboard shows a sortable position number ([pull-12](#))

Version 1.19 (2012-09-17)

- Prevent XSS in user name ([JENKINS-14309](#))
- authenticated users can modify their own scores ([JENKINS-3575](#))
- CI game fails with NPE ([JENKINS-11041](#))
- Added support for Markup Formatter in the LeaderBoard for displaying the user's description ([pull-2](#))

Version 1.18 (2011-02-18)

- added support for incremental Maven builds (not yet implemented for the PMD and the Violations rule)

Version 1.17

- Added stylecop warnings for the violations plugin [JENKINS-4847](#)

Version 1.16

- Fix issue with last analysis plugins [JENKINS-5022](#)

Version 1.15

- Fixed a bug where a player that exists in several SCM log entries would receive points for each entry. ([JENKINS-4470](#))

Version 1.14

- Fixed so skip unit tests are ignored. ([JENKINS-4449](#))

Version 1.13 *Requires Hudson 1.320 or later*

- Fixed so no points are awarded for a build if previous build reports has an error. ([JENKINS-4391](#))
- Hopefully fixed the NPE in the Violations plugin. ([JENKINS-3726](#))
- Added feature to handle user names case insensitive. ([JENKINS-3990](#))
- Fixed 404 when clicking on Leaderboard in any view but the main view ([JENKINS-3604](#))
- Added localisation support. So if you want help out, please check the [Internationalization](#) page. ([JENKINS-3484](#))

Version 1.12

- Added a description column in the leader board ([JENKINS-4141](#))
- Fixed most rules so the person who fixes a broken build doesn't get any penalties or rewards for it. The person will still get the points for fixing the build. ([JENKINS-2592](#))

Version 1.11

- Fixed so points are added or subtracted for test cases when run in a maven job ([JENKINS-2542](#))

Version 1.7

- Fixed NoSuchMethod exception when using newer versions of Warnings and Checkstyle plugins ([JENKINS-2169](#), [JENKINS-2218](#))

Version 1.6

- Points are no longer awarded for violations and checkstyle 'fixes' when the is build is broken ([JENKINS-1884](#))

Version 1.5

- Warnings, FindBugs, PMD, Open tasks, Checkstyle rules no longer throws exception when it is used for the first time ([JENKINS-1810](#))
- Anonymous users no longer see the "Reset scores" link next to the leader board ([JENKINS-1837](#))

Version 1.4

- Added option to exclude users from the game ([JENKINS-1608](#))
- Added option for Administrators to change the score of each user ([JENKINS-1608](#))
- Now it is possible for Administrators (or everyone if no there is no security) to reset all scores ([JENKINS-1754](#))

Version 1.3

- Using Violations plugin 0.6.0 no longer throws an exception ([JENKINS-1755](#))

Version 1.2 *Requires Hudson 1.211 or later*

- Fixed that points are given for each added/failed unit test ([JENKINS-1721](#))
- Added rules for the [PMD Plugin](#). Adding/removing a HIGH priority warning = -5/+5. Adding/removing a MEDIUM priority warning = -3/+3. Adding/removing a LOW priority warning = -1/+1.
- Added rules for the [Task Scanner Plugin](#). Adding/removing a HIGH priority task = -5/+5. Adding/removing a MEDIUM priority task = -3/+3. Adding/removing a LOW priority task = -1/+1
- Added rules for the [Violations Plugin](#). Adding/removing a violation = -1/+1. Adding/removing a duplication violation = +5/-5.
- Added rules for the [Findbugs Plugin](#). Adding/removing a HIGH priority findbugs warning = -5/+5. Adding/removing a MEDIUM priority findbugs warning = -3/+3. Adding/removing a LOW priority findbugs warning = -1/+1 ([JENKINS-1743](#))
- [Warnings Plugin](#). Adding/removing a compiler warning = -1/+1.
- [Checkstyle Plugin](#). Adding/removing a checkstyle warning = -1/+1.

Version 1.1

- Fixed that no points are given out when a build goes from failed to unstable

Version 1.0

- Initial version with hard coded rules