

JMH Report Plugin

Visually explore your [JMH](#) benchmarks on [Jenkins](#)!

Plugin Information

View JMH Report [on the plugin site](#) for more information.

Features

- Visualize the results of your [JMH](#) benchmarks via embedded [jmh-visualizer](#)
 - Project page report
 - Run page report

Changelog

- 0.9 On pair with [jmh-visualizer-0.9.1](#)
 - <https://github.com/jzillmann/jmh-visualizer/milestone/6?closed=1>
 - Support for Job DSL
- 0.8 On pair with [jmh-visualizer-0.8](#)
 - Revamped Summary page
- 0.7 On pair with [jmh-visualizer-0.7](#)
 - Support multiple runs in project view
- 0.6 On pair with [jmh-visualizer-0.6](#)
- 0.5.1 Add support for folder jobs
- 0.5 On pair with [jmh-visualizer-0.5](#)
- 0.4.3.3 Support for using JMH-Report on pipeline projects (Upgraded minimum Jenkins version from 2.0 to 2.4)
- 0.4.3.1 Fix erroneous context path handling which prevented the proper load of the reports
- 0.4.2 Initial Release

[Version Archive](#)



The plugin makes use of [jmh-visualizer project](#). The version number correspond to its releases. So [jmh-report-plugin 0.4.2](#) makes use of [jmh-visualizer 0.4.2](#).

Requirements

- Jenkins 2.4 or newer is required.
- The project you want the reporting on should:
 - Execute the JMH benchmarks
 - Store the benchmarks results in JSON format in any folder

How to's

Setup in pipeline project

You can setup [jmh-report](#) as a pipeline step:

```
jmhReport 'build/reports/jmh/result.json'
```

OR

```
step([{$class: 'RunPublisher', resultPath: 'build/reports/jmh/result.json'}])
```

A complete pipeline could look as simple as that:

```
node {
  checkout([$class: 'GitSCM', branches: [[name: '*/master']], doGenerateSubmoduleConfigurations: false,
  extensions: [], submoduleCfg: [], userRemoteConfigs: [[url: 'https://github.com/ben-manes/caffeine.git']]])
  sh './gradlew jmh -PincludePattern=".*DelegationBenchmark.*"'
  jmhReport 'caffeine/build/reports/jmh/results.json'
}
```

Setup with Job DSL

```
jmhReport { resultPath('build/reports/jmh/result.json') }
```

A complete job-dsl example could look like that:

```
job('example-1') {
  steps {
    scm {
      git("https://github.com/ben-manes/caffeine.git", "master")
    }
    shell('./gradlew jmh -PincludePattern=".*DelegationBenchmark.*"')
    publishers {
      jmhReport {
        resultPath('build/reports/jmh/result.json')
      }
    }
  }
}
```

Setup in regular project

You can setup jmh-report - i.e. in a freestyle project - by adding it as a post-build action.

Post-build Actions

Post-build Actions

JMH Report

JMH result path

Add post-build action ▼