

# Recording Unit Test Results

## Overview

Pipelines can record and display test result files that have been output by a unit test runner. Tests results data must be XML files in junit format for Pipelines to be able to display them.

When a pipeline has saved test results, the report can be viewed in the **Tests** tab for the step in the [Pipeline Run Logs](#).



## Page Contents

- [Overview](#)
- [Saving Test Results](#)
- [Example Pipeline](#)

## Saving Test Results

To record unit test results to the pipeline's run log:

- Run unit tests in a pipeline step that produce a results file in junit format
- Save the test results file using the [save\\_tests utility function](#)

```
save_tests $res_myGitRepo_resourcePath/testreport.xml
```

## Example Pipeline

The following example pipeline demonstrates the process of generating unit test results in a Maven build and recording the results to the run log for later viewing in Pipelines.

The pipeline defines a [GitRepo](#) resource that contains the Java application, and a [BuildInfo](#) resource for the resulting build.

### pipelines.resources.yml

```
resources:  
  - name: java_app  
    type: GitRepo  
    configuration:  
      gitProvider: MyGithub  
      path: myproject/simple-maven-sample  
      branches:  
        include: master  
  
  - name: my_build_info  
    type: BuildInfo  
    configuration:  
      sourceArtifactory: myArtifactory  
      buildName: simple-maven-sample  
      buildNumber: 1
```

The pipeline executes a Maven build using the [MvnBuild](#) native step. It requires that the Maven project being built has been configured to use the [Maven Surefire Plugin](#) to perform unit tests and produce a junit-compatible test result file.

In the `onComplete` action of the `execution` block, the step will save the test result file to the run log by calling the `save_tests` utility function.

## pipelines.steps.yml

```
pipelines:
- name: java_pipeline
  steps:
  - name: build_app
    type: MvnBuild
    configuration:
      sourceLocation: .
      mvnCommand: clean install -ntp
      configFileLocation: .
      configFileName: mvn-art-config
      autoPublishBuildInfo: true
      integrations:
        - name: myArtifactory
      inputResources:
        - name: java_app
      outputResources:
        - name: my_build_info
      runtime:
        type: image
        image:
          auto:
            language: java
            versions:
              - "11"
    execution:
      onComplete:
        - save_tests $res_java_app_resourcePath/target/surefire-reports/
```

After the pipeline has finished running, the test results for the `build_app` step can be viewed in the run log.

The screenshot displays the Jenkins interface for the `build_app` step. At the top, a dropdown menu shows `build_app`. To its right, a green checkmark indicates `Success`, followed by the step type `MvnBuild`, the start time `Started Last Tuesday at 12:33 PM`, and the duration `Ran for 3 minutes`. Below this, a navigation bar includes `Logs`, `Tests` (highlighted in green), `Step Info`, and `Resources`. The main content area shows a summary of test results: `✓ 3 Passing`, `! 0 Failing`, `▲ 0 Errors`, and `⏏ 2 Skipped`.