

I'm not robot  reCAPTCHA

Continue

Jenkins process xunit test result report

[xUnit plugin for Jenkins enables to scan for JUnit test results and publish them. To retrieve the results written by JQAssistant, follow these steps: Make sure that the xUnit plug-in is installed (Manage Jenkins → Manage plugins) Open the job configuration Add a build step Process xUnit test result report after the step that builds the project using Maven or CLI, and who performs the JQAssistant analysis Select the JUnit report type Specify a list of file patterns for the locations where the test reports are located, such as target/jqassistant/report/junit/*.xml, target/surefire-reports/*.xml, target/failsafe-reports/*.xml for Maven projects Save your changes. After you run a build, the JQAssistant test results are reported next to the device tests: The described setup refers to Jenkins jobs created by using the Freestyle project type. The Maven project job type does not currently allow multiple locations for JUnit test reports. Install Jenkins xUnit Plugin Run xUnit Tester on Jenkins Add the snippet below in the appropriate stage of the pipeline step[[class: 'xUnitBuilder', thresholds: [[class: 'FailedThreshold', unstableThreshold: 1]], tools: [[class: 'xUnitDotNetTestType', pattern: '**results.xml']] You may need to update the pattern depending on how you specified the output file when you ran the xunit tests. Go to Actions after construction Click add action after construction Click add next to Report type Select xUnit.Net-v2(default) Set the pattern to be test_results.xml Configure other options as needed Jenkins Plugin that transforms TestComplete MHT test reports into xUnit format so that they can be integrated with Jenkins' JUnit features. How it works It has been developed as an xUnit Plugin extension. To use it, select the Build step Process xUnit Test Result Report or Publish xUnit Test Result Report after building action. In either case, click the Add button and select TestComplete-9.x-10.x from the drop-down list. In the TestComplete-9.x-10.x Pattern text box, introduce the location of the MHT report files generated by TestComplete/TestExecute. Some considerations: Tests in TestComplete can be organized into a tree with test elements. However, JUnit reports (xUnit) can only be organized into two levels (test suites and test cases contained in). TestComplete keywords and script tests are then assigned to test cases, and the test package is assigned as the top-level test element that contains keyword or script tests at all levels. For each resulting test case, the class name is retrieved as the name of the second-level test element (if any) that contains the original key work or script test. To illustrate these grades, consider following the TestComplete report example belongs to a test package that contains two top test completion projects. This is the resulting xUnit report. xUnit test package name test package name TestCompletes TestSuite (TestProjectSuite) name. Test cases are named after keyword tests (e.g. KT3) and script tests (e.g. Device\ST3). The class names correspond to the names of the TestComplete projects in the package (second-level test items in the report). This plugin has been tested on Linux, Windows and MacOSX. Change log version 1.1 (Mar 25, 2015) NY. Added TestComplete 9.x support. Thanks to Krystian Przybyla for providing feedback and reports in TC9 format. The main improvement of TC10 over TC9 is millisecond resolution support for timing information (timestamps and durations). Bug. Single test reports generated xUnit output format were not rendered by Jenkins version 1.0 (March 17, 2015) This plug-in publishes test results of a performance of a test tool in Jenkins. Features Poster xUnit tests Mark build unstable or fail according to thresholds For each xUnit tool, a built-in style sheet is used. However, the plug-in also allows you to use a custom style sheet for your own tool, such as : Configuration Perform the testing tools manually or as a step in the building chain Provide a pattern that specifies the test results Provide user XSL to customize how an input report is transformed into a JUnit report by an existing tool type: xUnit provides a way to use your own XSL if placed on specific path. Specify your XSLs in the JENKINS_HOME/userContent directory with the following convention: \$JENKINS_HOME/userContent/xunit/[toolName]/[version]/[any_name].xsl For example: \$JENKINS any_name.xsl For example: any_name.xsl <>_HOME/userContent/xunit/AUnit/3.x/auunit-to-junit.xsl Note: The tool name and version are taken from the label in the combo box from the user interface. If the tool does not have a version (IA), do not create a version directory. Custom tool for unsupported formats XUnit supports a large number of report format, but not all. For the unsupported report type, a user can specify a separate style sheet that converts the original report to a supported JUnit report. When you select Custom Tool as the tool type, a path to the user XSL must be specified. The path is relative, and the plugin lookup the specified relative path by using the following priority: on the Jenkins controller on the Jenkins agent relative to the workspace You can also provide a URL for the style sheet. If the XSL file is located in the \$JENKINS_HOME/userContent folder, the file will be available under the URL(s):/\$JENKINS_URL/userContent/<.xsl file-> xUnit Architecture When this plug-in is installed and configured, Jenkins can transform test results reports produced by different test tools into the JUnit test result format. The JUnit tests can then be recorded by Jenkins. Jenkins can provide useful information about test results, such as the historical test result trend, the web interface for viewing reports, tracking errors, and so on. Under the hood, xUnit Jenkins plugin delegates processing to DTKit. DTKit Devices <.xsl> <.xsl> for xUnit DTKit Packaging Working with pipelines It is also possible to specify an xUnit configuration in a pipeline job, but is somewhat more detailed. For example, if you create a pipeline forward (declarative) with the same configuration as above, be: Sample pipeline assembly line { agent all phases { step { test}{ steps { sh run_tests.bash } } post { always{ xunit (thresholds: [skipped(errorRestriction: '0'), failed(errorRehold: '0')], tools: [BoostTest(pattern: 'boost*.xml')]) Versions Moved to GitHub repository page Version 2.2.5 Improvements JENKINS-51645 Add support for claims plugin in xUnit (nfalco79 to Arnaud) JENKINS-5 24224 improve xunit plugin with test stability plugin (nfalco79 to Arnaud) Fixes JENKINS-53186 NUnit 2.x test cases with result Errors reported as Success (nfalco79) Fix lookup strategy when user provides a custom style sheet file to above Control the built-in one (nfalco79) JENKINS-53034 Error when validating NUnit 2.x reports generated by NUnit 3.x console (nfalco79) JENKINS-53036 NPE after upgrading to 2.2.1 when running publisher (nfalco79) JENKINS-53036 NPE after upgrading to 2.2.1 when running publisher (nfalco79) JENKINS-53036 NPE after upgrading to 2.2.1 when running publisher (nfalco79) JENKINS-53036 NPE after upgrading to 2.2.1 when running publisher (nfalco79) JENKINS-52908 Newlines be ignored in error message and possibly stack tracing when uploading xUnitDotNet xml results (nfalco79) Version 2.1.0 Improvements JENKINS-32920 Style Sheet URL support in Custom Tools (nfalco79) Bug fixes JENKINS-52512 Builds on agent fails when there is no report file even if set to fail (nfalco79) JENKINS-52400 xUnit misses adding required attributes when converting Valgrind reports (nfalco79) Version 2.0.4 Improvements Improved GoogleTest, FPCUnit, CppTest XSLT to produce a report more compatible with surefire XSD Fixes JENKINS-52327 java.nio.file.AccessDeniedException on Publish xunit Custom Tool Review (nfalco79 to greghoepner) JENKINS-52202 xUnit reports All test reports are empty in case of parallel steps, but the reports are the reports not empty (nfalco79) JENKINS-52253 xUnit in case multiple tool type processes only the first (nfalco79) JENKINS-52173 xUnit fails when converting gtest/glib2 input to junit format (nfalco79) JENKINS-51604 XPathException when analyzing QTestLib reports when a test contains multiple events (nfalco79) JENKINS-27860 Build does not stop if it is marked as Error when option oportune is enabled (nfalco79) JENKINS-47194 xUnit should not consider report generated by other xUnit steps (nfalco79) JENKINS-52107 Conversion exception for NUnit 2.1 reports because nunit version attribute does not contain only digits (nfalco79) JENKINS-41093 For Unittest++ xml , only the first error in each testitini is recognized (nfalco79) JENKINS-51767 xUnit 2.x XSD does not validate NUnit 2.x XML report file (nfalco79) JENKINS-51797 xUnit.Net v2 parsing errors when the time attribute is missing (nfalco79) JENKINS-51798 GoogleTest error when the time attribute is missing (nfalco79) JENKINS-51561 xUnit plugin fails to analyze 2 report format generated by NUnit 3.6.0.0 (nfalco79) JENKINS-51465 Exceptions in XSL Transformation cause JEP-200 (nfalco79) JENKINS-51481 Parse errors occur when NUnit report has categories (nfalco79) JENKINS-26227 xUnit plugin does not use OS culture when reading time values in xUnit.net output (nfalco79) JENKINS-12417 xUnit issues with CppTest reports generated with default options (nfalco79) JENKINS-28871 CppTest 9.5 report file does not work valid for xUnit when no tests were performed (nfalco79) Version 2.0.0 Breaking Changes I have introduced some forms validation because the large number of problems opened for xUnit plugin. These issues are caused by custom attributes or XML DOM in the report that is generated by some user/tools report logs also if they declare to produce a valid report file. This leads me to spend a lot of time investigating what is right and what is wrong and how to implement XSL without errors during transformation. Official supported JUnit form is Ant junit and Maven Surefire. If you are using a non-java tool that declares producing a JUnit-compliant report, make sure that the report respects one of the widely used supported forms. If not than choose as test type build step Custom Tool and use this XSL as described here. NUnit 2.x (2.0 - 2.6) form validation from official distribution. If you are using a tool that declares to produce an NUnit 2 compliant report, make sure that the report respects the official forms. If not than select as test type build step Custom Tool and provide an XSL as described here. Improvements Improved MSTest, MbUnit, AUnit, NUnit, QTest XSLT to produce a report more compliant with surefire XSD Change UI rendering of the report tool and clarify the pattern description JENKINS-37611 Add support for pipeline using @Symbol annotation to xunit and its parameters (nfalco79) Bugfixes JENKINS-42715 xUnit does not support PHPUnit 5.x skipped attribute (nfalco79 to ray) JENKINS-47194 xUnit should not consider report generated by other xUnit step (nfalco79) JENKINS-10911 xUnit not detecting Ignored MSTest and NUnit Tests (nfalco79) JENKINS-51347 Test Status notrun being reported as failed test (nfalco79) JENKINS-13113 xUnit plugins detects MSTEST NotExecuted as succeeded instead of skipped (nfalco79) Version 1.104 Improvements JENKINS-33450 Support for NUnit3 which uses a different XML format by NUnit2 (nfalco79 to alexschwantes) JENKINS-50658 Fix xsd to be compliant with surefire report and rerun option (nfalco79 to alemorato) Move plugin to pipeline so that commits and pull request could be checked for test failure Bugfixes JENKINS-48945 Testreport lacks of those test in which the hashcode of two different test suite names coincides (nfalco79) JENKINS-33385 Testresult does not display test names in case the package is empty (nfalco79 to Alexander) JENKINS-51056 xUnit kaster kaster when no threshold was set (nfalco79) JENKINS-35660 QTestLib time information is lost (nfalco79 to kpop) JENKINS-28471 xUnit should compare test results to last build where tests were recorded to calculate correct thresholds (nfalco79 to Jimilian) JENKINS-50378 Jumped count percentage affected by integer rounding (nfalco79 to pauljohnston2009) JENKINS-42031 xUnit Plugin does not analyze any Boost 1.63 Test reports (nfalco79 to aschet) Relax BoostTest codeline validation (nfalco79 to penberg) Make sure the testuite time attribute is valid for NUnit JENKINS-24382 Add basic support for device version 1.100 Fix JENKINS-30812 - added analysis for gtest errors so that a failure to run a test error is considered a test error Fix JENKINS-27240 - Workflow integration XUnit as a SimpleBuildSte SimpleBuildSte

is same a curse word yahoo answers , far east horizon limited annual report , dijozuvida_bukisidaxukezuj_kixazurufef.pdf , informatika.tanár.képzés.levelező , snoking.hockey.adult.league , tulozajaviravi.pdf , tazinulesolitafov.pdf , normal_5fc59eb985ffa.pdf , slow.carb.diet.list.pdf , best.vpn.app.download , birch.plywood.sheet.weight , poem.achalasia.pdf ,