

MSTest Plugin

Plugin Information

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

This plugin converts [MSTest](#) TRX test reports into JUnit XML reports so it can be integrated with Jenkin's JUnit features. This plugin converts the `coveragexml` files found in the project workspace to the EMMA format.

You can use [MSTestRunner plugin](#) or [VsTestRunner plugin](#) to run the test and use this plugin to process the results.

The MSTest plugin analyzes the test execution reports (TRX) files generated by `mstest` and `vstest.console`. These files include a test execution summary, and detailed data about what happened during the tests execution. Along with the test execution records, `vstest` or `mstest` may also include a link towards the code coverage data collected during the tests execution. These coverage files have a binary, proprietary format, and you have to convert them to XML before that the MSTest publisher is invoked.

This plugin converts the test records to the JUnit format, and adds them to the build report. The plugin also searches for the link to the code coverage data. If it is present, and it contains valuable data, the plugin converts it to the Emma format by means of an ad-hoc XSL transformation.

If you want to show the coverage data on the build report, it is up to you to add the Emma plugin to your Jenkins instance, and to add the corresponding post-build action to your build. The coverage reports will be somewhere in the build workspace, and their name will match the pattern: ``emma\coverage.xml``. Apart from the Emma plugin, which is rather outdated, probably also other plugins can complete the same task (for example, the JaCoCo plugin).

Pipeline Support

The MSTest plugin supports the pipeline plugin (and/or Jenkinsfile build definitions). The plugin call be called with an instruction like

```
step([$class: 'MSTestPublisher', testResultsFile:"**/*.trx", failOnError: true, keepLongStdio: true])
```

Or with its shortest alternative:

```
mstest testResultsFile:"**/*.trx", keepLongStdio: true
```

Code Coverage Support

Since the code coverage data has a binary, proprietary format, and that the tools capable of handling them are released under a proprietary license along with the development environments, you will have to perform the data conversion yourself. Here's how.

Code Coverage Data Conversion

To convert the binary VSTest.Console output to the [Microsoft CoverageDS XML format](#), you may use one of the prebuilt applications referenced in the V0.14 release notes below, or you can build the following converter application:

CoverageConverter.exe

```
class Program
{
    static int Main(string[] args)
    {
        if ( args.Length != 2)
        {
            Console.WriteLine("Coverage Convert - reads VStest binary code coverage data, and outputs it in XML format.");
            Console.WriteLine("Usage:  CoverageConvert <sourcefile> <destinationfile>");
            return 1;
        }

        CoverageInfo info;
        string path;
        try
        {
            path = System.IO.Path.GetDirectoryName(args[0]);
            info = CoverageInfo.CreateFromFile(args[0], new string[] { path }, new string[] { });
        }
        catch (Exception e)
        {
            Console.WriteLine("Error opening coverage data: {0}",e.Message);
            return 1;
        }

        CoverageDS data = info.BuildDataSet();

        try
        {
            data.WriteXml(args[1]);
        }
        catch (Exception e)
        {
            Console.WriteLine("Error writing to output file: {0}", e.Message);
            return 1;
        }

        return 0;
    }
}
```

The CoverageDS and CoverageInfo types are being exposed by the Microsoft.VisualStudio.Coverage.Analysis.dll (official documentation: <https://msdn.microsoft.com/en-us/library/microsoft.visualstudio.coverage.analysis.coverageds.aspx>, article explaining how to use these assemblies with a nice step-by-step: <https://blogs.msdn.microsoft.com/phuene/2009/12/01/programmatic-coverage-analysis-in-visual-studio-2010/>)

The following Powershell build step will find one binary coverage data file in your workspace and convert it to XML format, assuming you use the default TestResults directory. (Modify as necessary to handle multiple coverage files)

Powershell build step for CoverageDS XML Conversion

```
$generatedCoverageFile = $(get-ChildItem -Path .\TestResults -Recurse -Include *coverage)[0]
CoverageConverter $generatedCoverageFile TestResults\vstest.coveragexml
```

Can I use Microsoft's CodeCoverage.exe for data conversion?

Microsoft supplies *C:\Program Files (x86)\Microsoft Visual Studio 1X.0\Team Tools\Dynamic Code Coverage Tools\CodeCoverage.exe* with Visual Studio. But unfortunately this tool uses an XML format distinct from the CoverageDS format used by vstest and mstest, and is not compatible with the MSTest plugin. If you are interested in this topic, and you've some experience with XSL, there are a couple of transforms to convert back and forth in the [github repository](#).

Change Log

Version 0.20 (September 1st, 2017)

- The release description is available on github: <https://github.com/jenkinsci/mstest-plugin/releases/tag/mstest-0.20>

Version 0.19 (September 1st, 2015)

- Support for web tests (contacted by email, by Peter Barnes. No Jira issue has been opened)
- Let the users still using Java 1.6 to continue using the plugin [\[JENKINS-29032\]](#)
- Mark the inconclusive tests as 'skipped' [\[JENKINS-29316\]](#)

Version 0.18 (May 12th, 2015) --- !!! Java 1.7 is required !!!

- Add support for "Retain long standard output/error" [\[JENKINS-28281\]](#). The default value for this option is false. If you're automating the creation of your jobs, simply specify `keepLongStdio=on` as a parameter of your query. Any other value than 'on' will set this option to false.
- Add localized messages for it, pt-BR, fr
- Cumulated code coverage filename: `vstest.coveragexml`
- Add default values for test result pattern and "fail if no result file is found": `*/*.trx` and `true`.

Version 0.17 (May 4th, 2015) --- !!! Java 1.7 is required !!!

- Add a checkbox to ignore missing TRX files (Thanks Christopher Bush, pull request [#7](#)). The pull request contains also a way to automate job creation using the REST API. So, if you're automating the creation of your jobs, just specify `failOnError=on` to enable this feature. Any other value than 'on' will set this option to false.
- Fix the code coverage calculations (Thanks junshanxu, pull request [#6](#)): a sum over all the nodes is better than using the value of the first node only.

Version 0.16 (Apr 14th, 2015) --- !!! Java 1.7 is required !!!

- Show the code coverage graph for `coveragexml` files (one of the two XSD, the one produced by `vstest`)

Version 0.15 (Apr 14th, 2015) --- !!! Java 1.7 is required !!!

- Improve support for data driven tests (Thanks, Darryl Melander: pull request [#6](#))
- Preserve charsets while fixing TRX files (JENKINS-23531, reopened by JitinJohn@MS)

Version 0.14 (Apr 1st, 2015)

- Support for output/stdout messages (JENKINS-19384)
- Drop invalid XML entities (JENKINS-23531). MSTest allows writing XML entities corresponding to invalid XML characters. These XML entities generate exceptions while being parsed by Java parsers. For me, it's still unclear if such entities are standard or not. However, to avoid these exceptions, the `mstest` parser simply drops them. These entities normally correspond to non printable characters.
- Support for `.coveragexml` files. The coverage data present in these files is being transformed in an EMMA coverage report. Today, you can try to generate `vscoveragexml` files using <https://github.com/gredman/CI.MSBuild.Tasks> or <https://github.com/yasu-s/CoverageConverter>.

Version 0.13 (Mar 18, 2015)

- Support for ignored tests (JENKINS_27469)
- Support for data driven tests (JENKINS-8193, JENKINS-4075)
- Support for timed out tests (JENKINS-11332)
- Support for TextMessages (JENKINS-17506)
- Improved processing for tests whose `@outcome` is not set
- Stacktraces are now shown as `stacktraces`, and error messages as `error messages`

Version 0.12 (Mar 12, 2015)

- Convert MS XML code coverage reports in emma coverage reports, and show them.
- Fix: the tests for which the outcome is 'error' (or missing, with an error message or a stack trace) will be reported as junit errors.

Version 0.11 (Jan 17, 2015)

- Support `vstest` TRX format
- Support environment variables as target (`vstestrunner-plugin` exports the full path to the TRX as environment variable)

Version 0.7 (Jun 17, 2011)

- Supported MSTest 2010 ordered tests (JENKINS-7458)
- Supported wildcard (JENKINS-8520)

Version 0.6 (Feb 11, 2010)

- Fixed issue [JENKINS-3906](#): Durations greater than 59s
- Fixed issue [JENKINS-4632](#): MSTest plugin does not parse Visual Studio 2010 results

Version 0.5 (Feb 6, 2010)

- Update code for more recent Hudson

Version 0.4 (Jun 16, 2009)

- Fixed the *AbortException* issue
- Added i18n support
- Added Brazilian portuguese localization

Version 0.3

- Identifies test's class using the ExecutionId variable

Version 0.2

- Fixed a problem to identify namespace and class name from the TestMethod tag
- Changed JUnit test report file name

Version 0.1

- Initial Release