

PikeTec TPT Plugin

Plugin Information

View piketec-tpt [on the plugin site](#) for more information.

TPT is a testing and verification tool for embedded control systems. TPT can test MATLAB Simulink or dSPACE TargetLink models, ETAS ASCET models, C-Code or test via MiL, SiL, PiL and HiL. The PikeTec TPT Plugin allows users to execute tests modeled in [TPT \(Time Partition Testing\)](#) via Jenkins. A XML file can be generated in JUnit format for the reporting of test results. You can also display the TPT test report in Jenkins.

Usage

We will give you a short overview here. If you need a more detailed description please refer to the chapter *Jenkins Continuous Integration* of TPT manual.

The plugin provides two build steps:

1. *Execute TPT test cases*: The TPT test cases are executed and the test results are converted into JUnit-XML files. If Jenkins is operating as master in master-slave-mode, it will delegate the work to slave jobs.
2. *Execute TPT tests slave*: Execute the work delegated to it by *Execute TPT test cases* in master slave mode.

You have two options to publish the TPT test results:

1. Add a *TPT Report* post build action to your job to display the TPT report in Jenkins.
2. Configure the build steps to transform the test results into the JUnit format and then publish the results in Jenkins using the *Publish JUnit test result report* post build action provided by the [JUnit Plugin](#).

Using only *Execute TPT test cases* (not master slave mode)

Simply create a new Jenkins Job and add the *Execute TPT test cases* build step and configure it as needed. Do not set the option "Distribute work to TPT slave jobs".

Make sure:

- The TPT software is installed on the computers where tests should be run.
- All 3rd party tools needed on the Jenkins node where the job will run are available.
- The correct TPT project file is available (use e.g. SCM checkout to check this).
- The JUnit XML directory is placed in the workspace of the job in case you want to publish the test results to Jenkins.

Using only *Execute TPT test cases* and *Execute TPT tests slave* (master slave mode)

You can configure the *Execute TPT test cases* build step to delegate the work to a slave job by setting the option "Distribute work to TPT slave jobs". The slave job must contain a *Execute TPT tests slave* build step.

The *Execute TPT test cases* build step will open the specified TPT file, lookup the tests, split them in packages with an equal number of tests and starts for each package a build of the slave job.

When all slave jobs are finished the *Execute TPT test cases* build step will copy the test data into its workspace. The job will now generate an overall report and if you configured the job for publishing via JUnit the JUnit XML files.

Make sure:

- The job containing the *Execute TPT test cases* build step runs on the master.
- Only one slave job build runs the node at any given time.

The *Execute TPT tests slave* will communicate with TPT via a network protocol (Java RMI). If two builds are using the same port and would run on the same node the outcome of the execution would be nondeterministic. Use e.g. the [Throttle Concurrent Builds Plugin](#) to prevent builds from running on the same node.

FAQ

TPT hangs/does not start when I use Jenkins. If I start TPT normally it works fine.

This is usually a problem with the licensing. Since Jenkins normally runs as a service with a different user TPT will access other settings than when executed manually. If you have a correctly configured TPT you can copy the license configuration file "%LOCALAPPDATA%\TPT\$\InstallDirName_hash\$\license_serverconfig.cfg" (please replace \$InstallDirName_hash\$ by your TPT installation directory name and the hash) into the TPT installation directory and rename it to "license_default.cfg". If TPT does not find a license configuration it will use these instead.

Please note that other 3rd party tools as MATLAB Simulink or ETAS ASCET may have their own issues running in a service environment.

I cannot publish the TPT test results. The JUnit publisher cannot find anything.

The JUnit publisher can only find files in the workspace of the Job while the *Execute TPT test cases* allows you to create the XMLfiles anywhere in the file system. Please ensure you configured the paths correctly. You can use a relative path. A valid configuration would be "junit" as the path given to the build step and "junit/*.xml" given to the publisher.

Release History

Version 8.6

- Last minute changes of TPT 13 API added.

Version 8.5

- Lifted TPT API version to TPT 13.
- If you use master slave mode and have multiple execution configurations added to a TPT build step only the last test results were tranformed to JUnit xml.

Version 8.4

- Lifted TPT API version to TPT 12.

Version 8.3

- Null pointer exception fixed if runnig older projects that have not seen the test set feature introduced in 8.0.
- Retry for slave jobs will now work again.

Version 8.2

- TPT Report publisher now works on Linux nodes
- TPT Report publisher is now compatible with folder plugin
- TPT Report publisher now finds testdata created by TPT build steps embedded in other steps (e.g. conditional build step)

Version 8.1

- Fixed a Bug that prevented tables of new publisher from showing any content

Version 8.0

- Publisher for TPT reports added
- Removed dependencies to other plugins: Copy-to-slave and Parametrized Trigger
- It is now possible to explicitly set the test set to execute.

Version 7.8

- It is now possible to reduce the TPT log entries that are added to failed JUnit tests by setting a severitiy level.

Version 7.7

- Retry logic was inverted: Successful jobs were scheduled again and faild jobs not.
- If a TPT execution configuration the multi core feature enabled the execution with master and slave jobs failed because the testdata dir coul not be cleaned
- The TPT file is now closed on the master after execution, too.

Version 7.6

- A slave job can know accept more than one TPT test case and it is now possible to define the number of slave jobs started. The test cases will be distributed evenly to all started slave jobs so you can use the TPT multicore feature running multiple test cases in paralell in every TPT instance.
- If a slave job execution fails (red ball) it is now possible to configure the master job to reschedule it up to the specified number of times.

Version 7.5

- Variables in tpt file field are now expanded as well.

Version 7.4

- When upgrading from a version prior to 6.5 the "Path to tpt.exe" configuration was lost and a null pointer exception occurred. Fix now works for newer Jenkins versions.

Version 7.3

- When upgrading from a version prior to 6.5 the "Path to tpt.exe" configuration was lost and a null pointer exception occurred.

Version 7.1

- After execution TPT projects are now closed on master and slave nodes.
- If a test set of an execution configuration entry does not contain the test case to execute the entry will be temporarily deactivated.
- If a TPT project cannot be opened the job fails and the errors are dumped to the Jenkins console.

Version 7.0

- Added master-slave functionality for distributing TPT test workload.

Version 6.7

- Execution on Slave-Nodes could fail because existence check and creation of necessary folders were always executed on Master-Node.

Version 6.6

- A 6h timeout was hardcoded for TPT to finish. The timeout is now configurable.

Version 6.5

- The Field "Path to tpt.exe" now accepts a comma seperated list of paths. The first existing one is used for testexecution.

Version 6.4

- Variables are now expanded so you can use them in the configuration fields
- Help files are now accessable again
- more robust against misplaced quotes
- least needed Jenkins version set to LTS 1.625.1

Version 6.3

- Initial public release