Build Failure Conditions

In TeamCity you can adjust the conditions when a build should be marked as failed in the Failure Conditions section of the of the Build Configuration Settings page.

On this page:

- Common build failure conditions
- Additional Failure Conditions
    - Fail build on metric change
        - Adding custom build metric
    - Fail build on specific text in build log

Common build failure conditions

In the Common Failure Conditions, specify how TeamCity will fail builds by selecting appropriate options from in the Fail build if area:

- it runs longer than ... minutes: Check this option and enter a value in minutes to enable execution timeout for the build. If the specified amount of time is exceeded, the build is automatically canceled. This option helps to deal with builds that hang and maintains agent efficiency.
- the build process exit code is not zero: Check this option to mark the build as failed if the build process doesn't exit successfully.
- at least one test failed: Check this option to mark the build as failed if the build fails at least one test. If this option is not checked, the build can be marked successful even if it fails to pass a number of tests. Regardless of this option, TeamCity will run all build steps.
- an error message is logged by build runner: Check this option to mark the build as failed if the build runner reports an error while building.
- an out of memory or crash is detected (Java only): Check this option to mark the build as failed if a crash of the JVM is detected, or Java out of memory problems. If possible, TeamCity will upload crash logs and memory dumps as artifacts for such builds.

Additional Failure Conditions

You can instruct TeamCity to mark a build as failed if some of its metrics has deteriorated in comparison with another build,e.g. code coverage, artifacts size, etc. For instance, you can mark build as failed if the code duplicates number is higher than in the previous build.

Another build failure condition causes TeamCity to mark build as failed when a certain text is present in the build log.

To add such build failure condition to your build configuration, click Add build failure condition and select from the list:

- Fail build on metric change
- Fail build on specific text in build log

> ✓ You can disable a build failure condition temporarily or permanently at any time, even if it is inherited from a build configuration template.

Fail build on metric change

When using code examining tools in your build, like code coverage, duplicates finders, inspections and so on, your build generates various numeric metrics. For these metrics you can specify a threshold which, when exceeded, will fail a build.

> In general there are two ways to configure this build fail condition:
>
> - A build metric exceeds or is less than the specified threshold. For example: Fail build if `build duration (secs)` compared to constant value is `more` than* 300. In this case a build will fail if it runs more than 300 seconds.
> - A build metric has changed comparing to a specific build by a specified value. For example: Fail build if its `build duration (secs)` compared to a value from another build is `more` by at least `300` default units for this metric than the value in the `Last successful build`. In this case a build will fail if it runs 300 seconds longer than the last successful build. If Branch specification is configured, then the following logic is applied.

The following builds can be used as the basis for comparing build metics:

- last successful build

- last pinned build
- last finished build
- build with specified build number
- last finished build with specified tag.

By default, TeamCity provides the wide range of build metrics:

- artifacts size(bytes)
- build duration (secs)
- number of classes
- number of code duplicated
- number of covered classes
- number of covered lines
- number of covered methods
- number of failed tests
- number of ignored tests
- number of inspection errors
- number of inspection warnings
- number of lines of code
- number of methods
- number of passed tests
- number of tests
- percentage of block coverage
- percentage of class coverage
- percentage of line coverage
- percentage of method coverage
- percentage of statement coverage
- test duration (secs)
- total artifacts size (bytes)

⚠ Note that since TeamCity 9.0, the way TeamCity counts tests has changed.

Adding custom build metric

You can add your own build metric. To do so, you need to modify the TeamCity configuration file `<TeamCity Data Directory>/config/main-config.xml` and add the following section under "server" node there:

```
<build-metrics>
<statisticValue key="myMetric" description="build metric for number of files"/>
</build-metrics>
```

So, if your build publishes the `myMetric` value, you can use it as a criterion for a build failure.

Fail build on specific text in build log

TeamCity can inspect all lines in build log for some particular text occurrence that indicates a build failure.
Lines are matched without the time and block name prefixes which precede each message in the build log representation.

To configure this build failure condition, specify:

- a string or a Java Regular Expression whose presence/absence in the build log is an indicator of a build failure,
- a failure message to be displayed on the build results page when a build fails due to this condition.