

FAQs and Troubleshooting

In this section:

- [How can I get quick access to information about usage on the command line?](#)
- [How can I prevent my machine ID from floating?](#)
- [What if Jtest reports compilation problems when building the project with Maven?](#)
- [Why does running the `jtest:coverage` goal \(Maven\) or `jtest-coverage` task \(Gradle\) fail the build?](#)
- [What if Jtest cannot collect coverage information for tests run with Gradle?](#)
- [How can I work with Jtest via proxy?](#)
- [Why are suppressions of some rules no longer available on DTP after Jtest was upgraded to a newer version?](#)

How can I get quick access to information about usage on the command line?

Use the `-help` command line switch:

```
jtestcli.exe -help
```

How can I prevent my machine ID from floating?

Changes in the network environment may affect the interface that is used to compute your machine ID and result in machine ID instability. You can use the `PARASOFT_SUPPORT_NET_INTERFACES` environment variable to specify a stable interface and prevent the machine ID from floating.

1. Set up the `PARASOFT_SUPPORT_NET_INTERFACES` environment variable.
2. Set the variable value to a stable Ethernet network interface. Do not use virtual, temporary or loopback interfaces.
 - On Windows: Set the value to the MAC address of your network card. You can use the `ipconfig -all` command to obtain the address. For example:

```
SET PARASOFT_SUPPORT_NET_INTERFACES=00-10-D9-27-AC-85
```

- On Linux: Set the value to one of the network interfaces from the "inet" or "inet6" family. You can use the `ifconfig` command to obtain the list of available interfaces. For example:

```
export PARASOFT_SUPPORT_NET_INTERFACES=eth1
```

If the problem persists, you can obtain diagnostic information by setting up the environment variable `PARASOFT_DEBUG_NET_INTERFACES` and setting its value to true. This will print to the standard output the checking procedure that can be shared with technical support, as well as the interface that is used to compute your machine ID. The interface will be marked with the [SELECTED] prefix.

What if Jtest reports compilation problems when building the project with Maven?

If Jtest reports compilation problems related to importing dependencies in the Maven test scope, try running the `jtest:jtest` goal with the `mvn test` command:

```
mvn test jtest:jtest
```

Why does running the `jtest:coverage` goal (Maven) or `jtest-coverage` task (Gradle) fail the build?

The Jtest `coverage` goal/task was dropped with release 10.2.2 (plugin version 1.2.4) and is no longer available (see [Migration From 10.x to 10.2.2 or later](#)). Executing this goal fails the build with the following messages:

```
[ERROR] Could not find goal 'coverage' in plugin com.parasoft.jtest:jtest-maven-plugin. (Maven)
```

```
Task 'jtest-coverage' not found (Gradle)
```

What if Jtest cannot collect coverage information for tests run with Gradle?

To enable collecting coverage, Jtest automatically configures the JVM arguments for the Jtest coverage agent. Overriding these arguments prevents Jtest from collecting coverage data. If your Gradle build script modifies JVM arguments, ensure they are added (+) to other JVM arguments to prevent overriding:

```
✘ jvmArgs = [  
  '--module-path', classpath.asPath,  
  '--add-modules', 'ALL-MODULE-PATH',  
  '--add-reads', "$moduleName=junit",  
  '--patch-module', "$moduleName=" + files(sourceSets.test.java.outputDir).  
  asPath  
]
```

```
✔ jvmArgs += [  
  '--module-path', classpath.asPath,  
  '--add-modules', 'ALL-MODULE-PATH',  
  '--add-reads', "$moduleName=junit",  
  '--patch-module', "$moduleName=" + files(sourceSets.test.java.outputDir).  
  asPath  
]
```

How can I work with Jtest via proxy?

Typically, if you connect through a proxy server, you need to configure the connection by passing protocol-specific system properties to the JVM – using the `-D` command line option.

To work with Jtest, ensure that the system properties for the HTTPS protocol (`https.proxyHost` and `https.proxyPort`) are configured. Your command line may resemble the following:

```
java -Dhttps.proxyHost=myserver.example.com -Dhttps.proxyPort=8080
```

i If you use Jtest on desktop with Eclipse or IntelliJ IDEA, the proxy settings are automatically detected and do not need to be configured in the command line.

Why are suppressions of some rules no longer available on DTP after Jtest was upgraded to a newer version?

Suppressions associated with rules whose messages changed between releases may not be available on DTP and the rules must be re-suppressed.

i You can restore legacy messages for BD category rules in version 10.4.1 and later by configuring the following settings in your `.properties` file:
`flowanalysis.legacy.messages.for.<rule_ID>=true.`

For example:

```
flowanalysis.legacy.messages.for.BD.PB.ARRAY=true  
flowanalysis.legacy.messages.for.BD.PB.ZERO=true
```