

Configure test environment

Test cases can be run as a regular JUnit test case using the JUnit *TestRunner* class or configuration tools such as Ant or Maven. However, some test case specific system properties should be set.

The program installation directory should be also specified using the *install.root* system property which can be passed to JVM as a runtime argument *-Dinstall.root=path_to_modeling_tool_installation_directory*.

It is possible to setup the program installation to force using the floating license server. The following properties should be passed to JVM as runtime arguments:

```
-DFL_FORCE_USAGE=true
-DFL_SERVER_ADDRESS=flexnet_license_server_address
-DFL_SERVER_PORT=license_server_port
-DFL_EDITION=magicdraw_edition
```

Test cases may use external resources such as configuration files and projects during tests. It is recommended to keep external test resources apart from test case implementation class files. The test framework provides methods for retrieving test resources from the one specific resource directory. The resource directory for test cases can be configured by specifying the *tests.resources* system property of JVM. The tests resource property can be passed to JVM as a runtime argument

```
-Dtests.resources=path_to_resources_directory
```

The Log4j logger used in test cases can be configured by specifying the *test.properties* file which should be placed in *<modeling tool installation directory>\data*.

The program is started in a "silent" mode when running tests - the user interference requiring dialogs are not shown, instead the warning "WARN GENERAL - TRYING TO DISPLAY "<dialog title>" DIALOG IN SILENT MODE" is logged. The stack trace is logged too if a special system property - *DprintSilentDialogStackTrace=true* is set.

Related Pages

- [Developing plugins using IDE](#)