

Creating test cases

Our modeling tools, including MagicDraw, provide a JUnit (www.junit.org)-based test frameworks. The main purpose of the test framework is to simplify the automatic unit and integration tests development for the program and its plugins. The test framework can be used by developers for testing their own plugins or standard essential program features.

JUnit 3 test cases

The test framework consists of an abstract JUnit test case implementation ([com.nomagic.magicdraw.tests.MagicDrawTestCase](#)) and a number of tools ([com.nomagic.magicdraw.tests.common.TestEnvironment](#), [com.nomagic.magicdraw.tests.common.comparators.ProjectsComparator](#)) that can be used for the following purposes:

- Starting the application
- Managing projects
- Checking program for memory leaks

JUnit 4 test cases

The test framework also has test class runner implementations ([com.nomagic.magicdraw.tests.MagicDrawTestRunner](#), [com.nomagic.magicdraw.tests.ParametrizedMagicDrawTestRunner](#)) that:

- Start the application
- Check program for memory leaks.

JUnit 5 test cases

Extend your test class with [com.nomagic.magicdraw.tests.MagicDrawApplication](#) to start the application before running the test methods.



You can find the examples of the code in `<modeling tool installation directory>\openapi\examples\testframework`

On this page

- [JUnit 3 test cases](#)
- [JUnit 4 test cases](#)
- [JUnit 5 test cases](#)

Related pages

- [Creating JUnit test case](#)
- [Comparing projects for testing purposes](#)
- [Working with test resources](#)
- [Configure test environment](#)