Taking advantage of NUnit XML reports

- About NUnit
- Nunit Basic Concepts
- Importing NUnit XML reports
  - Entities
    - Mapping of fields from the report to the Test issue
    - Examples
  - Status
- References

About NUnit

Nunit.org

NUnit is a testing framework for C# (C sharp), mostly focused on unit testing.

Similar to JUnit, NUnit is also used for writing integration and acceptance tests, making use of other libraries such as Selenium.

NUnit XML reports may be created by many different testing frameworks for C#, Javascript or any other language.

NUnit is quite feature-complete as may be seen in its online documentation.

Supported versions

Xray supports NUnit v2.6 and v3.0 XML reports.

Nunit Basic Concepts

In NUnit, you have Tests, Parameterized Tests, (Test) Suites and TestFixtures, among other things.

- A Test is a test case.
- Parameterized Tests are a way of specifying input values for a given Test (similar to an example in a Cucumber Scenario Outline).
- A Suite is a way of aggregating a group of tests for running.
- A TestFixture corresponds to a "class" containing multiple Tests.

NUnit uses "attributes" in order to ascribe behavior/characteristics to certain parts of your automated test code.

Learn in practice

Please look at the basic C# example: Testing using NUnit in C#.

Importing NUnit XML reports

Below is a simplified example of an NUnit 3.0 XML report containing a Test Suite with some Test Cases.

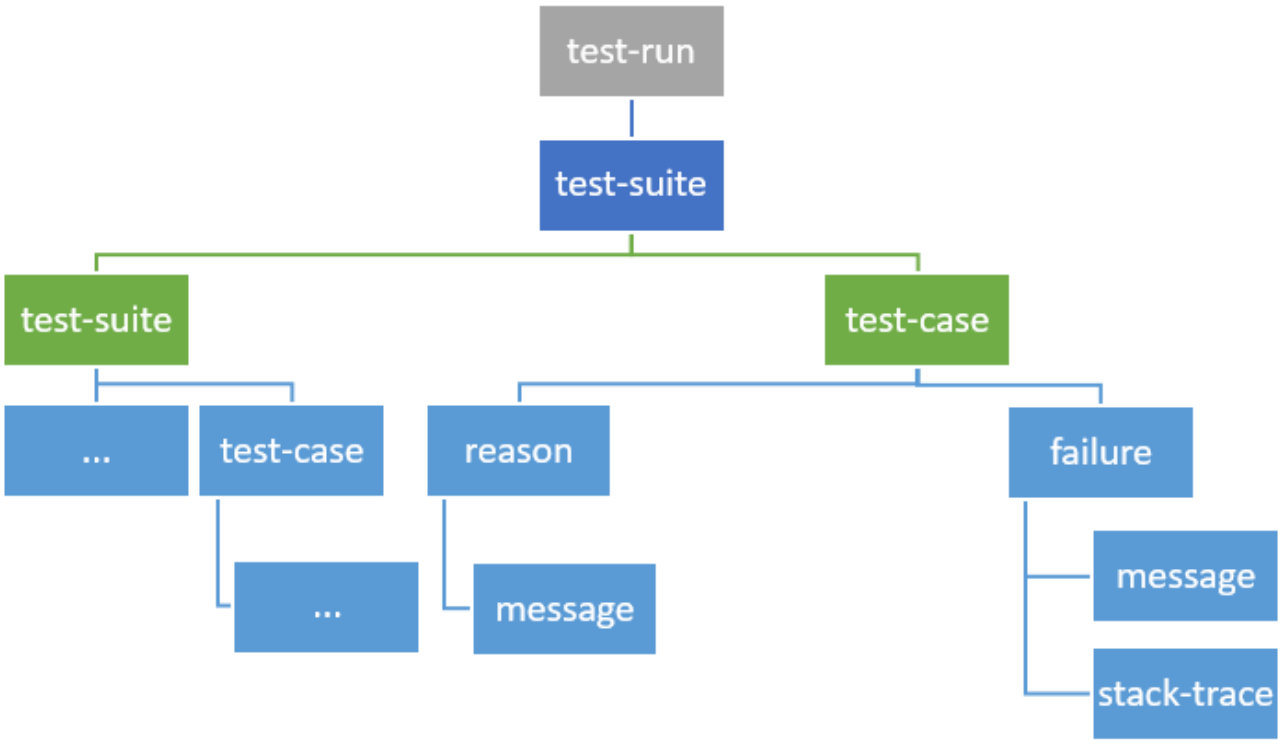
Sample Nunit 3.0 XML report

```
<?xml version="1.0" encoding="utf-8"?>
<test-run id="0" testcasecount="14" result="Failed" total="14" passed="13" failed="1" inconclusive="0" skipped="
0" asserts="14" portable-engine-version="3.3.0.0" start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:
40Z" duration="0.220000">
  <test-suite type="Assembly" id="1021" name="x, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null"
fullname="x, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" runstate="Runnable" testcasecount="14"
result="Failed" site="Child" start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="
0.096125" total="14" passed="13" failed="1" inconclusive="0" skipped="0" asserts="14">
    <settings>
      <setting name="WorkDirectory" value="C:\Users\Sergio\x" />
    </settings>
    <failure>
      <message><![CDATA[One or more child tests had errors]]></message>
```

```
    </failure>
    <test-suite type="TestFixture" id="1000" name="TestClass" fullname="TestClass" classname="TestClass"
runstate="Runnable" testcasecount="2" result="Failed" site="Child" start-time="2017-01-21 09:20:40Z" end-time="
2017-01-21 09:20:40Z" duration="0.072502" total="2" passed="1" failed="1" inconclusive="0" skipped="0" asserts="
2">
      <failure>
        <message><![CDATA[One or more child tests had errors]]></message>
      </failure>
      <test-suite type="ParameterizedMethod" id="1003" name="SubtractTest" fullname="TestClass.SubtractTest"
classname="TestClass" runstate="Runnable" testcasecount="2" result="Failed" site="Child" start-time="2017-01-21
09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.069346" total="2" passed="1" failed="1" inconclusive="0"
skipped="0" asserts="2">
        <failure>
          <message><![CDATA[One or more child tests had errors]]></message>
        </failure>
        <test-case id="1001" name="SubtractTest(1)" fullname="TestClass.SubtractTest(1)" methodname="
SubtractTest" classname="TestClass" runstate="Runnable" seed="366007487" result="Failed" start-time="2017-01-21
09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.037291" asserts="1">
          <failure>
            <message><![CDATA[  Expected: 10
  But was:  1
]]></message>
            <stack-trace><![CDATA[at TestClass.SubtractTest(Int32 x) in C:\Users\Sergio\x\TestClass.cs:line 13
]]></stack-trace>
          </failure>
        </test-case>
        <test-case id="1002" name="SubtractTest(10)" fullname="TestClass.SubtractTest(10)" methodname="
SubtractTest" classname="TestClass" runstate="Runnable" seed="718072923" result="Passed" start-time="2017-01-21
09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000087" asserts="1" />
      </test-suite>
    </test-suite>
    <test-suite type="TestSuite" id="1022" name="x" fullname="x" runstate="Runnable" testcasecount="12" result="
Passed" start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.014617" total="12"
passed="12" failed="0" inconclusive="0" skipped="0" asserts="12">
      <test-suite type="TestFixture" id="1004" name="CalculatorTests" fullname="x.CalculatorTests" classname="x.
CalculatorTests" runstate="Runnable" testcasecount="12" result="Passed" start-time="2017-01-21 09:20:40Z" end-
time="2017-01-21 09:20:40Z" duration="0.014413" total="12" passed="12" failed="0" inconclusive="0" skipped="0"
asserts="12">
        <test-suite type="ParameterizedMethod" id="1008" name="CanAddNumbers" fullname="x.CalculatorTests.
CanAddNumbers" classname="x.CalculatorTests" runstate="Runnable" testcasecount="3" result="Passed" start-time="
2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.003875" total="3" passed="3" failed="0"
inconclusive="0" skipped="0" asserts="3">
          <properties>
            <property name="Requirement" value="CALC-1" />
          </properties>
          <test-case id="1005" name="CanAddNumbers(1,1,2)" fullname="x.CalculatorTests.CanAddNumbers(1,1,2)"
methodname="CanAddNumbers" classname="x.CalculatorTests" runstate="Runnable" seed="294103071" result="Passed"
start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.001020" asserts="1" />
          <test-case id="1006" name="CanAddNumbers(-1,-1,-2)" fullname="x.CalculatorTests.CanAddNumbers(-1,-1,
-2)" methodname="CanAddNumbers" classname="x.CalculatorTests" runstate="Runnable" seed="1182014034" result="
Passed" start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000091" asserts="1" />
          <test-case id="1007" name="CanAddNumbers(100,5,105)" fullname="x.CalculatorTests.CanAddNumbers
(100,5,105)" methodname="CanAddNumbers" classname="x.CalculatorTests" runstate="Runnable" seed="1348027914"
result="Passed" start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000068" asserts="
1" />
        </test-suite>
        <test-suite type="ParameterizedMethod" id="1020" name="CanDivide" fullname="x.CalculatorTests.
CanDivide" classname="x.CalculatorTests" runstate="Runnable" testcasecount="3" result="Passed" start-time="2017-
01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.002578" total="3" passed="3" failed="0"
inconclusive="0" skipped="0" asserts="3">
          <test-case id="1017" name="CanDivide(1,1,1)" fullname="x.CalculatorTests.CanDivide(1,1,1)"
methodname="CanDivide" classname="x.CalculatorTests" runstate="Runnable" seed="1138926382" result="Passed"
start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000262" asserts="1" />
          <test-case id="1018" name="CanDivide(-1,-1,1)" fullname="x.CalculatorTests.CanDivide(-1,-1,1)"
methodname="CanDivide" classname="x.CalculatorTests" runstate="Runnable" seed="794915688" result="Passed" start-
time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000055" asserts="1" />
          <test-case id="1019" name="CanDivide(100,5,20)" fullname="x.CalculatorTests.CanDivide(100,5,20)"
methodname="CanDivide" classname="x.CalculatorTests" runstate="Runnable" seed="1794972673" result="Passed"
start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000048" asserts="1" />
        </test-suite>
        <test-suite type="ParameterizedMethod" id="1016" name="CanMultiply" fullname="x.CalculatorTests.
```

```
CanMultiply" classname="x.CalculatorTests" runstate="Runnable" testcasecount="3" result="Passed" start-time="
2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.002411" total="3" passed="3" failed="0"
inconclusive="0" skipped="0" asserts="3">
          <test-case id="1013" name="CanMultiply(1,1,1)" fullname="x.CalculatorTests.CanMultiply(1,1,1)"
methodname="CanMultiply" classname="x.CalculatorTests" runstate="Runnable" seed="343632280" result="Passed"
start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000300" asserts="1" />
          <test-case id="1014" name="CanMultiply(-1,-1,1)" fullname="x.CalculatorTests.CanMultiply(-1,-1,1)"
methodname="CanMultiply" classname="x.CalculatorTests" runstate="Runnable" seed="55985941" result="Passed"
start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000050" asserts="1" />
          <test-case id="1015" name="CanMultiply(100,5,500)" fullname="x.CalculatorTests.CanMultiply
(100,5,500)" methodname="CanMultiply" classname="x.CalculatorTests" runstate="Runnable" seed="1688307993"
result="Passed" start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000045" asserts="
1" />
        </test-suite>
        <test-suite type="ParameterizedMethod" id="1012" name="CanSubtract" fullname="x.CalculatorTests.
CanSubtract" classname="x.CalculatorTests" runstate="Runnable" testcasecount="3" result="Passed" start-time="
2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.002535" total="3" passed="3" failed="0"
inconclusive="0" skipped="0" asserts="3">
          <test-case id="1009" name="CanSubtract(1,1,0)" fullname="x.CalculatorTests.CanSubtract(1,1,0)"
methodname="CanSubtract" classname="x.CalculatorTests" runstate="Runnable" seed="1249440584" result="Passed"
start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000405" asserts="1" />
          <test-case id="1010" name="CanSubtract(-1,-1,0)" fullname="x.CalculatorTests.CanSubtract(-1,-1,0)"
methodname="CanSubtract" classname="x.CalculatorTests" runstate="Runnable" seed="1502662439" result="Passed"
start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000052" asserts="1" />
          <test-case id="1011" name="CanSubtract(100,5,95)" fullname="x.CalculatorTests.CanSubtract(100,5,95)"
methodname="CanSubtract" classname="x.CalculatorTests" runstate="Runnable" seed="1953557701" result="Passed"
start-time="2017-01-21 09:20:40Z" end-time="2017-01-21 09:20:40Z" duration="0.000047" asserts="1" />
        </test-suite>
      </test-suite>
    </test-suite>
  </test-suite>
</test-run>
```

The simplified tags hierarchy of these reports are shown in the following diagram:



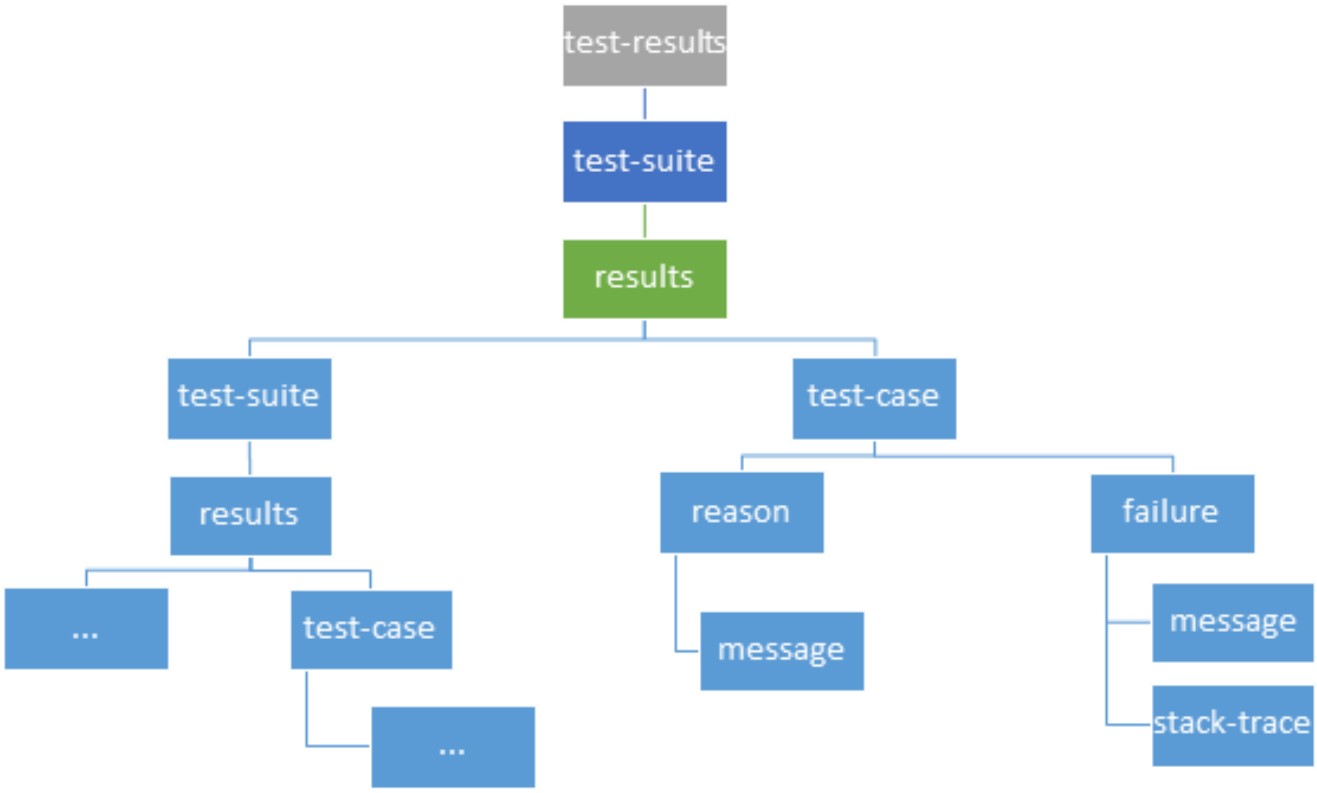
Below is an NUnit 2.6 XML report containing a Test Suite with some Test Cases.

Sample Nunit 2.6 XML report

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<test-results name="D:\Work\XpandIT\Repo\jm.customermobility\JMMobilityServices\JMMobilityServices.
ShoppingLists.Tests\bin\Debug\JMMobilityServices.ShoppingLists.Tests.dll" total="17" errors="0" failures="1"
not-run="0" inconclusive="0" ignored="0" skipped="0" invalid="0" date="2017-01-18" time="16:06:34">
  <environment nunit-version="2.6.4.14350" clr-version="2.0.50727.8745" os-version="Microsoft Windows NT
6.2.9200.0" platform="Win32NT" cwd="D:\Work\Apps\NUnit-2.6.4\bin" machine-name="LAPTOP-XPRT" user="rui.tome"
user-domain="LAPTOP-XPRT" />
  <culture-info current-culture="pt-PT" current-uiculture="en-GB" />
  <test-suite type="Assembly" name="D:\Work\XpandIT\Repo\jm.
customermobility\JMMobilityServices\JMMobilityServices.ShoppingLists.Tests\bin\Debug\JMMobilityServices.
ShoppingLists.Tests.dll" executed="True" result="Failure" success="False" time="57.360" asserts="0">
    <results>
      <test-suite type="Namespace" name="JMMobilityServices" executed="True" result="Failure" success="False"
time="57.355" asserts="0">
        <results>
          <test-suite type="Namespace" name="ShoppingLists" executed="True" result="Failure" success="False"
time="57.355" asserts="0">
            <results>
              <test-suite type="Namespace" name="Tests" executed="True" result="Failure" success="False" time="
57.355" asserts="0">
                <results>
                  <test-suite type="Namespace" name="Controllers" executed="True" result="Failure" success="
False" time="57.354" asserts="0">
                    <results>
                      <test-suite type="TestFixture" name="ShoppingListsControllerTests" executed="True"
result="Failure" success="False" time="57.354" asserts="0">
                        <results>
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.AddClient" executed="True" result="Success" success="True" time="4.519" asserts="
4" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.AddShoppingListItem" executed="True" result="Success" success="True" time="2.357"
asserts="1" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.CreateShoppingList" executed="True" result="Failure" success="False" time="0.341"
asserts="0">
                            <failure>
                              <message><![CDATA[]]></message>
                              <stack-trace><![CDATA[at JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.<CreateShoppingList>d__11.MoveNext() in D:\Work\XpandIT\Repo\jm.
customermobility\JMMobilityServices\JMMobilityServices.ShoppingLists.
Tests\Controllers\ShoppingListsControllerTests.cs:line 158
at NUnit.Framework.AsyncInvocationRegion.AsyncTaskInvocationRegion.WaitForPendingOperationsToComplete(Object
invocationResult)
at NUnit.Core.NUnitAsyncTestMethod.RunTestMethod()
]]></stack-trace>
                            </failure>
                          </test-case>
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.DeleteShoppingList" executed="True" result="Success" success="True" time="3.603"
asserts="2" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.GetPublicShoppingLists" executed="True" result="Success" success="True" time="
2.054" asserts="4" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.GetShoppingList" executed="True" result="Success" success="True" time="1.850"
asserts="1" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.GetShoppingLists" executed="True" result="Success" success="True" time="2.124"
asserts="2" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.GetSuggestedProducts" executed="True" result="Success" success="True" time="0.374"
asserts="0" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.RemoveAllShoppingListItems" executed="True" result="Success" success="True" time="
```

```
5.882" asserts="4" />
                        <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.RemoveBoughtShoppingListItems" executed="True" result="Success" success="True"
time="4.741" asserts="5" />
                        <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.RemoveClient" executed="True" result="Success" success="True" time="4.332"
asserts="2" />
                        <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.RemoveShoppingListItem" executed="True" result="Success" success="True" time="
2.742" asserts="2" />
                        <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.TestAuthorization" executed="True" result="Success" success="True" time="0.336"
asserts="1">
                          <properties>
                            <property name="Requirement" value="XPTO-15" />
                          </properties>
                        </test-case>
                        <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.UnmarkBoughtShoppingListItems" executed="True" result="Success" success="True"
time="6.842" asserts="5" />
                        <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.UpdateShoppingListItemBoughtStatus" executed="True" result="Success" success="
True" time="2.983" asserts="3" />
                        <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.UpdateShoppingListItemQuantity" executed="True" result="Success" success="True"
time="6.434" asserts="3" />
                        <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.UpdateShoppingListName" executed="True" result="Success" success="True" time="
3.148" asserts="2" />
                      </results>
                    </test-suite>
                  </results>
                </test-suite>
              </results>
            </test-suite>
          </results>
        </test-suite>
      </results>
    </test-suite>
  </results>
</test-suite>
</test-results>
```

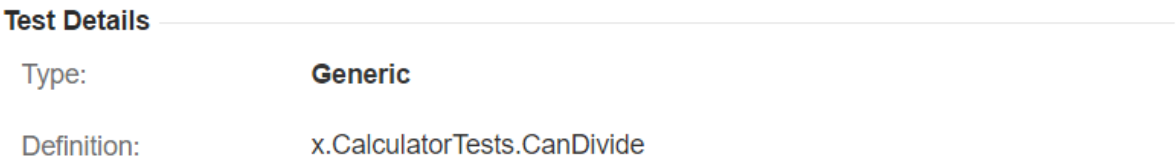
The simplified tags hierarchy of these reports are shown in the following diagram:



Entities

Test Cases are imported to Xray's ***Generic*** **Test issues.** For NUnit 3.0, the "classname" and "methodname" attributes are concatenated and mapped to the **Generic Test Definition** field of the Generic Test.

If a Test already exists with the same **Generic Test Definition**, then it is not created again.



Test Cases are imported to a new (or a user-specified) Test Execution in the context of some project, along with their respective execution results.

Parameterized Tests are imported to the same Test Run, where each set of parameters is identified by a different context.

NUnit's Test Suites are not mapped to any special entity. However, the execution details screen will show the Test Suite related to a specific test result.

Mapping of fields from the report to the Test issue

NUnit XML report	Test in Jira
• Nunit3.0: "methodname" attribute • Nunit2.6: last substring of "name" attribute (e.g. "xxx.xxx.xxx.CanDivide")	Summary field
• Nunit3.0: "classname" attribute + "." + "methodname" attribute • Nunit2.6: "name" attribute	Generic Test Definition custom field
categories	labels (except for the ones that match with Tests or requirements)

"Test" property or category containing a Jira key of a Test	identification of Test issue key in Jira
"Requirement" property or category containing a Jira key of a requirement	link to requirement

Notes:

- requirements may be identified by using either NUnit's categories or a property named "Requirement";
- Tests may be identified by using either Nunit's categories or a property named "Test";
  - if categories contain multiple references to Test entities, only the first instance is considered;
  - if the "Test" property is used explicitly, then the Test must exist or else it will not be imported.
- If the category references a requirement or a Test issue key, then the issue key should be represented with underscore instead of hyphen, e.g., "CALC_1" (due to the character limitation in NUnit).

Examples

Consider running some Tests implemented in the following C# class.

excerpt of C# TestFixture class

```
namespace x
{
    [TestFixture]
    public class CalculatorTests
    {
        [TestCase(1, 1, 2)]
        [TestCase(-1, -1, -2)]
        [TestCase(100, 5, 105)]
        [Category("CALC_1")]
        [Category("fast")]
        [Property("Requirement", "CALC-69")]
        [Property("Test", "CALC-10")]
        public void CanAddNumbers(int a, int b, int expected)
        {
            Assert.That(Calculator.Add(a, b), Is.EqualTo(expected));
        }


        [TestCase(1, 1, 0)]
        [TestCase(-1, -1, 0)]
        [TestCase(100, 5, 95)]
        [Category("CALC_2")]
        public void CanSubtract(int x, int y, int expected)
        {
            Assert.That(Calculator.Subtract(x, y), Is.EqualTo(expected));
        }
     ...

    }
}
```

In the first Test, Xray would try to find a Test with the key "CALC-10". If it does not exist, the Test won't be created.

The test would be linked to the Requirement with key "CALC-69", using the "tests" association, if the requirement exists.

If there is a requirement with the key "CALC-1", it would create the link to that requirement. If "CALC-1" corresponds to a Test issue, then "CALC_1" is ignored.

If the Category "CALC_1" does not map to either a requirement or a Test, it would be added to the Test as a label.

In the second Test, a Generic Test with the summary "CanSubtract" would be created.

The Test is only created if no Test with same Generic Test Definition already exists or if "CALC-2" does not correspond to a Test issue.

If there is a requirement with the key "CALC-2", it would create the link to that requirement.

If the Category "CALC_2" does not map to either a requirement or a Test, it would be added to the Test as a label.

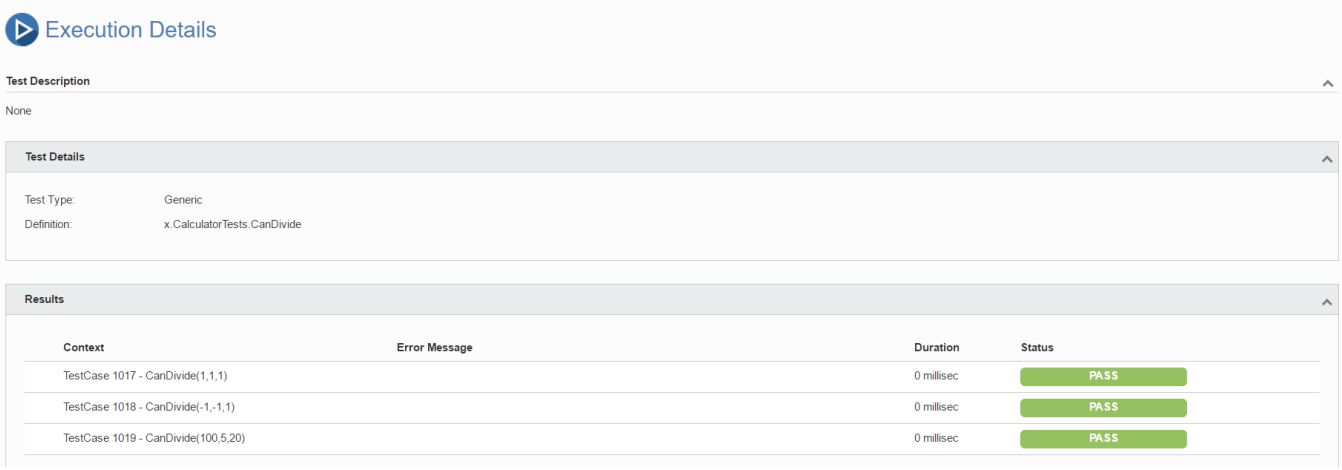
Status

The status of the Test Run will be set based on the Test case result:

Test Cases (NUnit3.0)	Test Cases (NUnit2.6)	Test status
Failed	**Error**	FAIL
Passed	**Success**	PASS
Other Cases	**Other Cases**	TODO

Note: Test Cases with the status FAIL may have a failure message displayed in the Test Run screen, under the Results section.

If the same Test Case has been executed in multiple Test Suites, then the result for each Test Suite will be shown. A parameterized Test will also appear with its own specific context.



Whenever a Test Case is executed in multiple Test Suites, the overall status of the Test Run will be calculated as a joint value.

Condition	Overall status of the Test Run
If all the mapped results of the Test Case was PASS	**PASS**
If any of the mapped results of the Test Case was FAIL	**FAIL**
Other cases	**TODO**

References

- http://www.nunit.org/
- http://nunit.sourceforge.net/doc/cookbook/cookbook.html
- https://github.com/nunit/docs/wiki/Test-Result-XML-Format