

# FREECOL

## Existing Test Cases

---

### Existing Test Cases

#### Introduction

FreeCol currently has 431 existing test cases, written using JUnit 3, which cover functionality in the client, server, common, and util packages. There are several helper and base classes for these tests such as `FreeColTestCase`, which adds some FreeCol-specific functionality to the `junit.framework` package's `TestCase` class, and `ServerTestHelper`, which provides stubbed server functionality. The entire test suite for the project has a locale setting which can be changed to run the tests in different locales.

#### Test Case #1: `ServerPlayer - testMarketRecovery`

This test case tests the behavior of the market after players buy and sell. It sets up two separate markets, a French market and an English market. It then sells lightly in the French market and heavily in the English market and checks to make sure that prices in the English market have dropped more drastically. Lastly, it advances time to check that the markets both eventually recover.

After studying this test case, we understood more about the expected behaviors of the “market” feature in the game, such as the fact that players’ markets are separate, that prices are expected to rise when products are sold in it, and that it recovers over time. Without the test case and included comments, we would not have been able to confirm the intent behind these features without playing the original game.

#### Test Case #2: `MonarchTest`

This test cases checks the monarch’s ability to raise or lower taxes given the minimum and maximum tax rates. The first test cases check for the monarch’s choices when taxes are set

---

---

at the minimum. If the tax is set at the minimum, then the monarch can only raise taxes through war or tax-acts. The monarch does not have the ability to lower them further. Similarly, when the taxes are set to maximum, the monarch can not raise taxes further and can only decrease taxes through war or tax-acts. Another test case checks for the default behavior where the tax is neither at its maximum or minimum, and the monarch has the ability to raise and/or lower taxes. Interestingly, the taxes are not applied to a player that is labeled a rebel.

After studying this test case, we learned that taxes have an impact on all players, except rebels, and that the monarch has limited taxing powers in relation to minimum and maximum tax rates.

### **Test Case #3: MissionAssignmentTest - testImpossibleConditionsForTargetSelection**

This test case tests the ability, or lack of ability, for one unit to attack another unit. By studying the test case, we can learn a couple of conditions that must be preset before an attack command can even occur.

The first unit is that a map, with a board, must exist, and the two pieces must be next to each other. We can also learn that players must be in a state of war before attacking. Players cannot target their own units to attack, and land units cannot attack naval units. If two nations who have the War stance target an enemy unit who are next to them, then an attack command is valid, and the game will process the outcome.