Testing with EiffelStudio

Christian Estler
ETH Zurich
christian.estler@inf.ethz.ch

Distributed and Outsourced Software Engineering - ETH course, Fall 2012

Information

There's a full tutorial about the following slides available at:

http://docs.eiffel.com/book/eiffelstudio/using-autotest

Why Testing?

We test in order to expose faults in our software

While testing will (most likely) not reveal **all** bugs, it can in increase the quality of the software significantly

Regression testing: changes to software in place A might break functionality in place B → a solid test-suite helps to find such bugs

In general: writing test cases takes time; tools try to help (e.g Junit for Java)

Testing and Contracts

Eiffel support Design-by-Contract

Contracts specify

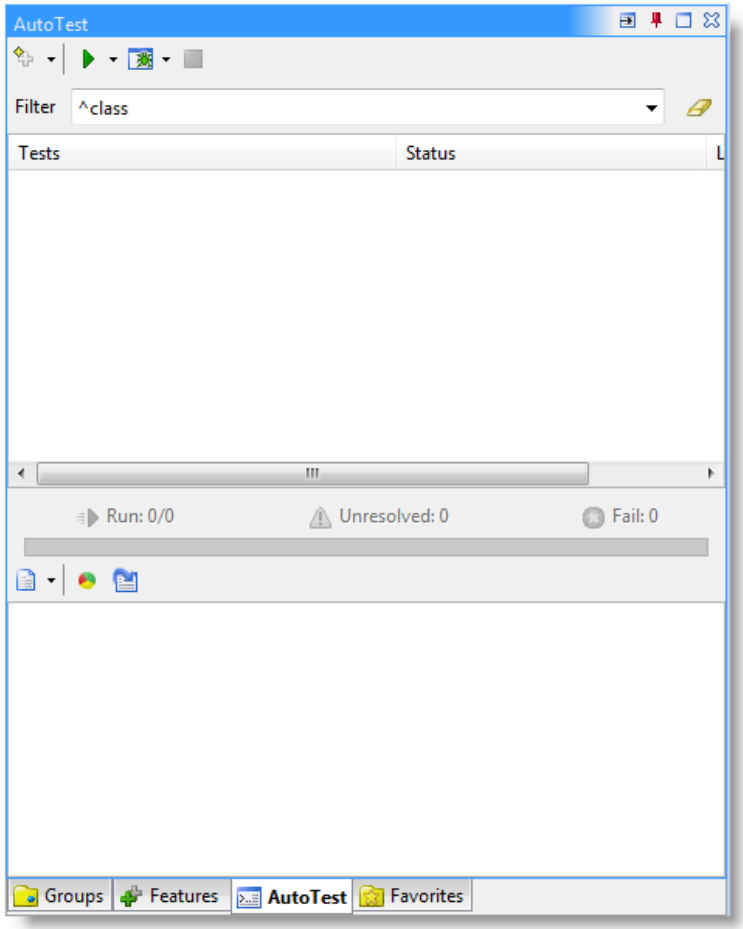
- What a routine expects (*precondition*)
- What a routine guarantees (*postcondition*)
- What an object maintains (*invariant*)

If code is annotated with contracts, we can test the code against it's specification → i.e. contracts can serve as a "oracles"

AutoTest

EiffelStudio has a built-in tool called **AutoTest**

- Simplifies creating, running and maintaining test
- Similar to JUnit test tool in Eclipse



AutoTest supports 3 types of tests

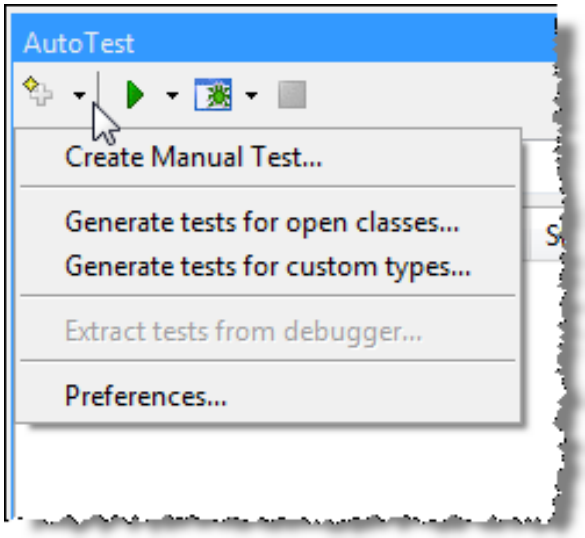
1. Manual tests
2. Extracted tests
3. Generated test

Generated tests are experimental, don't use them for DOSE

Manual Tests

A Wizard guides through the creation of test cases

- Creating a new manual test case
  - Goto the AutoTest tool
  - Click on the “+” symbol



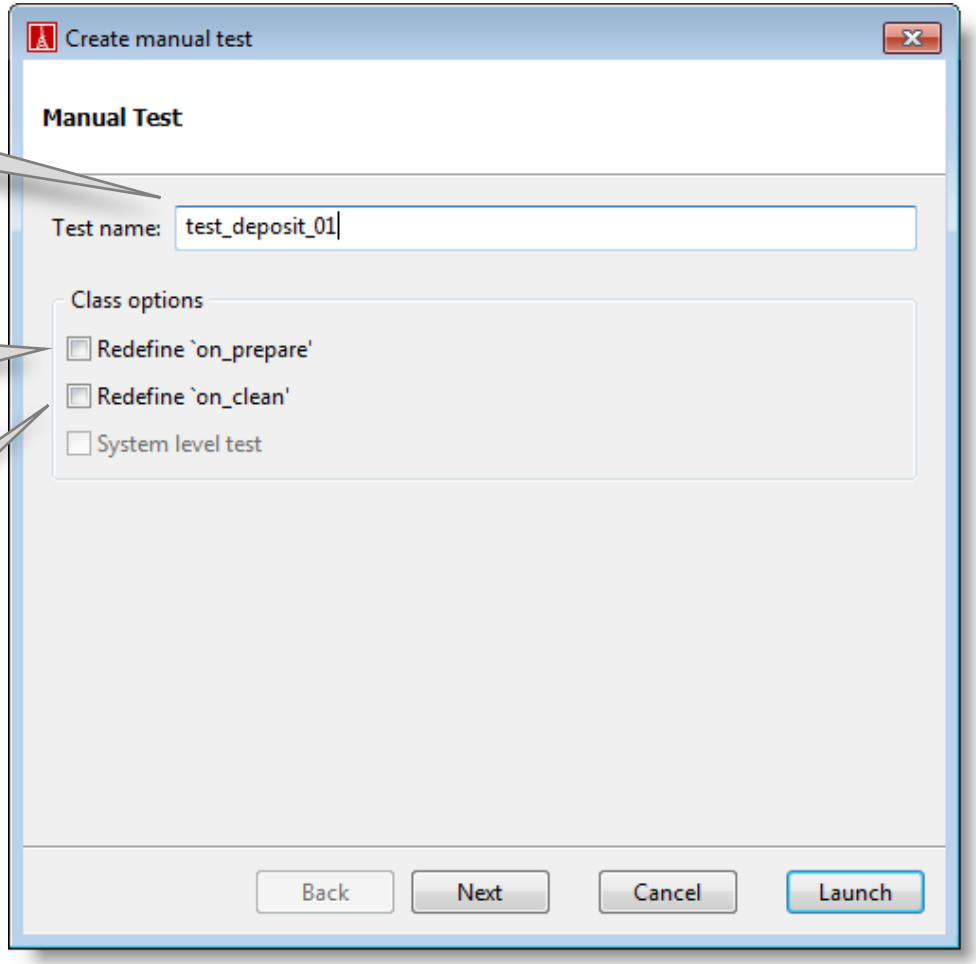
Manual Tests

The Manual Test Pane asks for the name of the test

Test name will be the name of the test-routine.

Before the test-case is run, a setup procedure is execute. We can add our own setup.

After executing the test case, a clean-up is performed. We can add our own clean-up.

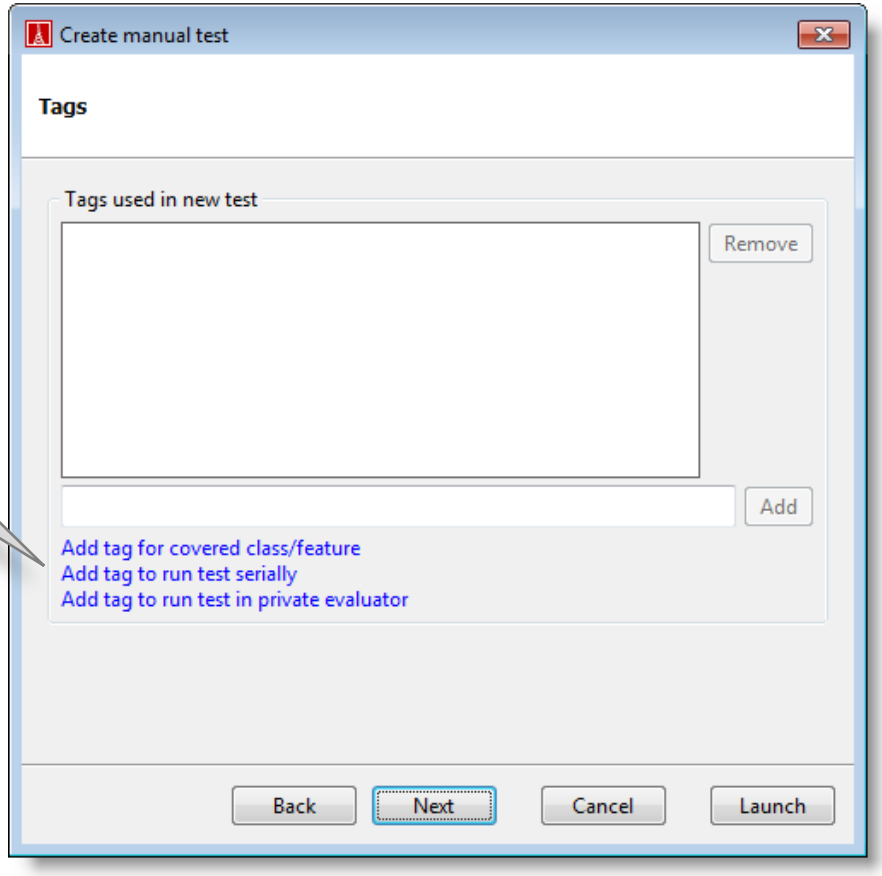


Manual Tests

Once we have many tests, it is useful to structure them.

That simplifies test-suit management and maintenance.

3 options to add tags to the test case



Details about tags:

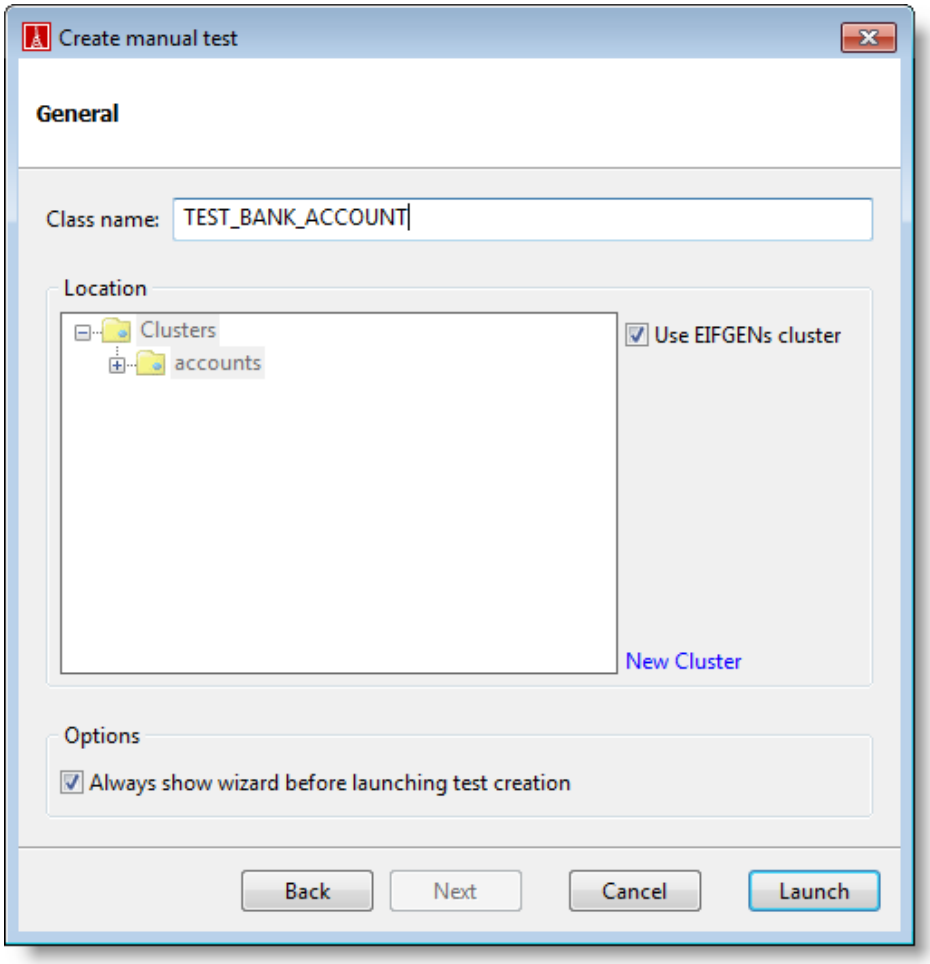
http://docs.eiffel.com/book/eiffelstudio/create-manual-test#About_Tags

Manual Tests

Finally, we define where the test case shall be stored

Tip:

Don’t use the EIFGENs folder. It’s the temporary folder that you delete from time to time (clean compile)



Manual Tests

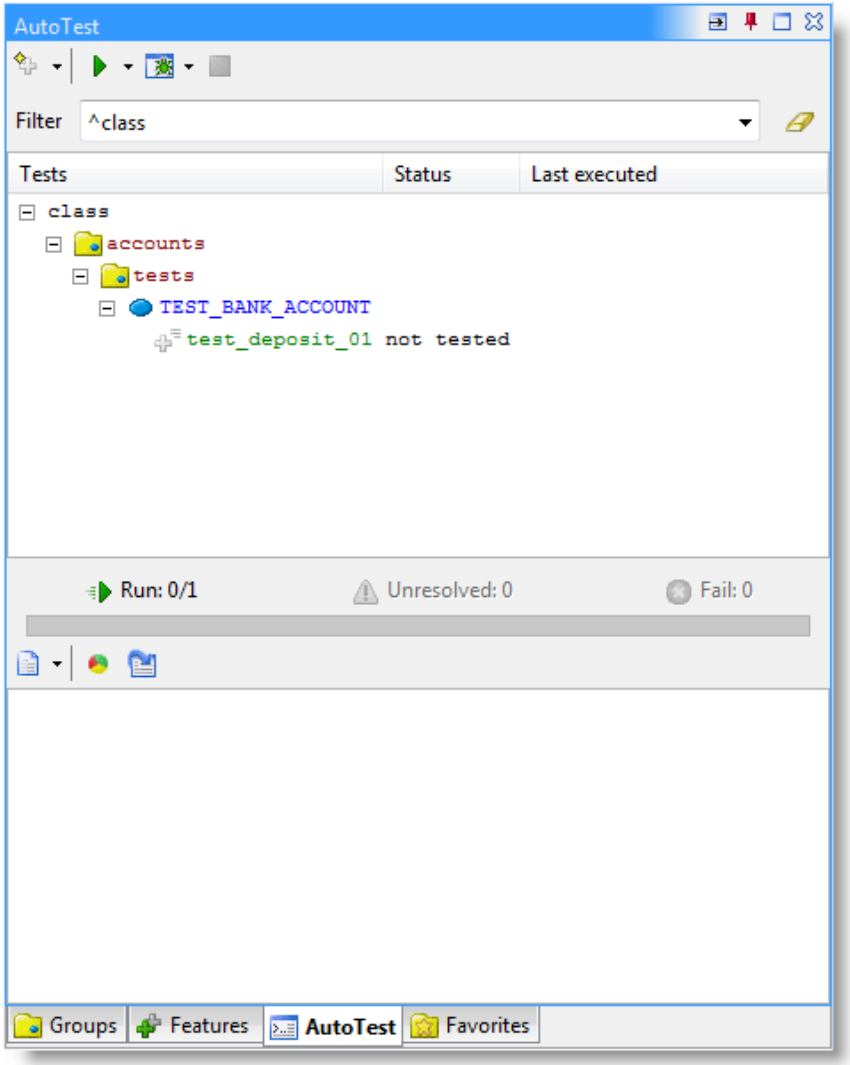
With clicking the “Launch” button we have created a test case

Next, we run the test case

Executing Tests

In the AutoTest tool

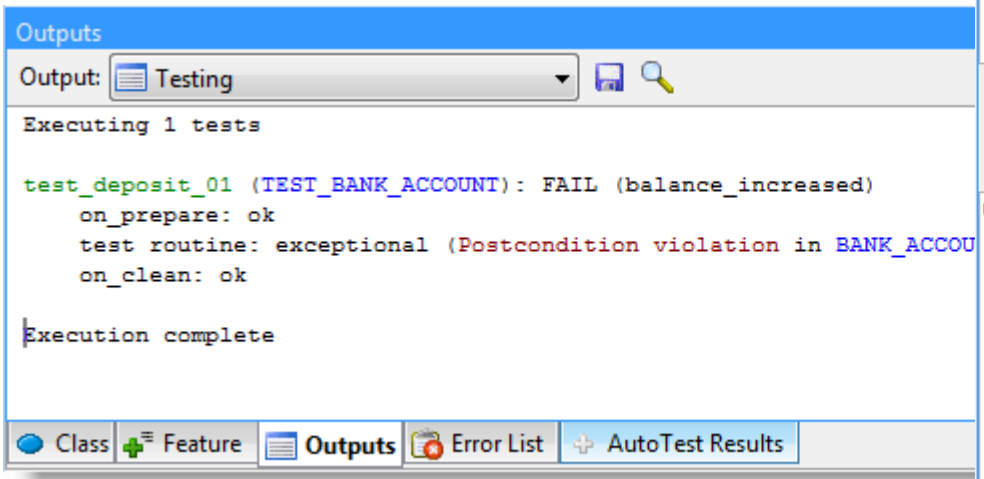
- Hit the “Run” Button
- That executes all selected tests
- If nothing is selected, all tests are executed

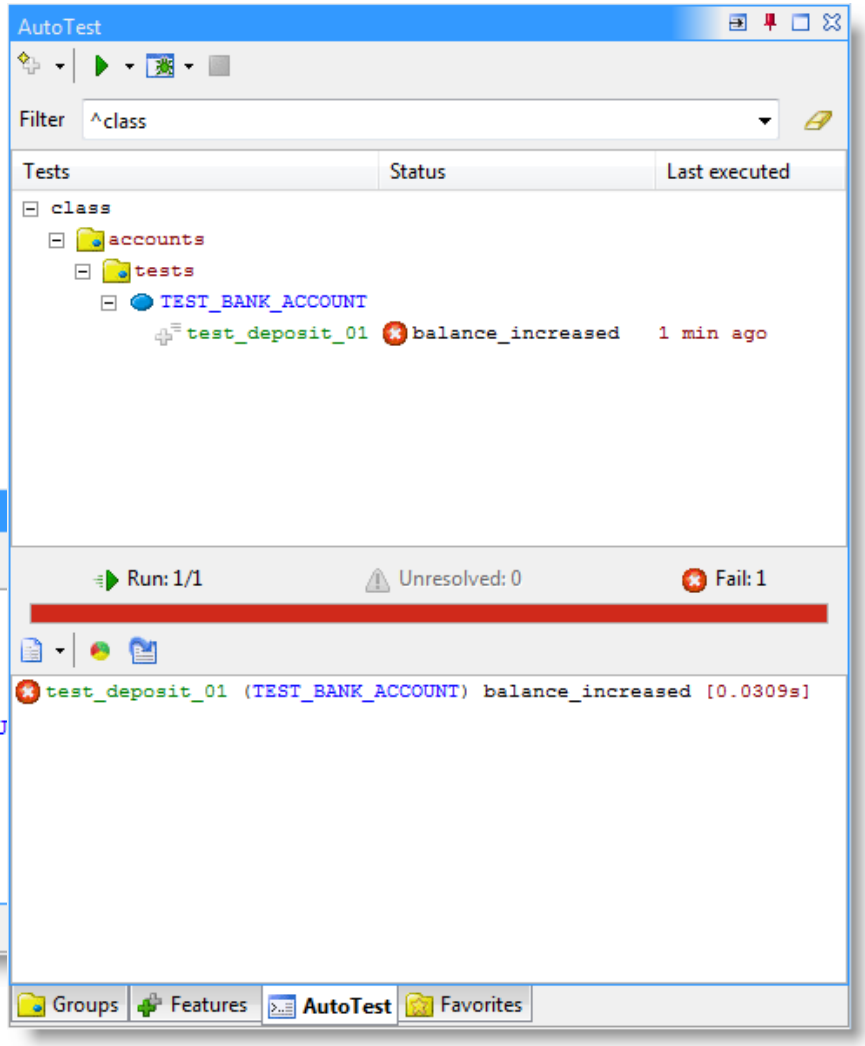


Executing Test

There are two result views

- Within AutoTest
- In the Output-Tool (under “Testing” Output)





Manual Tests - Oracles

We can the code against it's contracts

Additionally, we can use **`assert`** statements

- Structure: `assert (a_tag: STRING; a_condition: BOOLEAN)`

- Useful whenever the contract is not expressive enough/not testing the right thing

Extracted Tests

Whenever we run a debugging session and the application is paused, we can **extract** the current state into a test case

In particular useful for bug-fixing

Extracted Tests

DEMO

(no slides)

Summary

- DOSE
  - Invest into your group; don’t think locally
  - Try tools and methods; we won’t stop you
- EiffelStudio
  - Clean-Compile, Debugging
  - Don’t lose your sleep something breaks
- Agents & Tuples
- EiffelBase and EiffelVision
- AutoTest
  - Important & useful for phase 4 (TDD)

The End