



▶▶ How I learned to test without sleep

From coping to exploiting

A definition of testing

“Testing is the process of evaluating a product by learning about it through exploration and experimentation, which includes:
questioning, study, modeling,
observation and inference,
output checking,
etc.”

(Michael Bolton & James Bach)

Why "without sleep"?

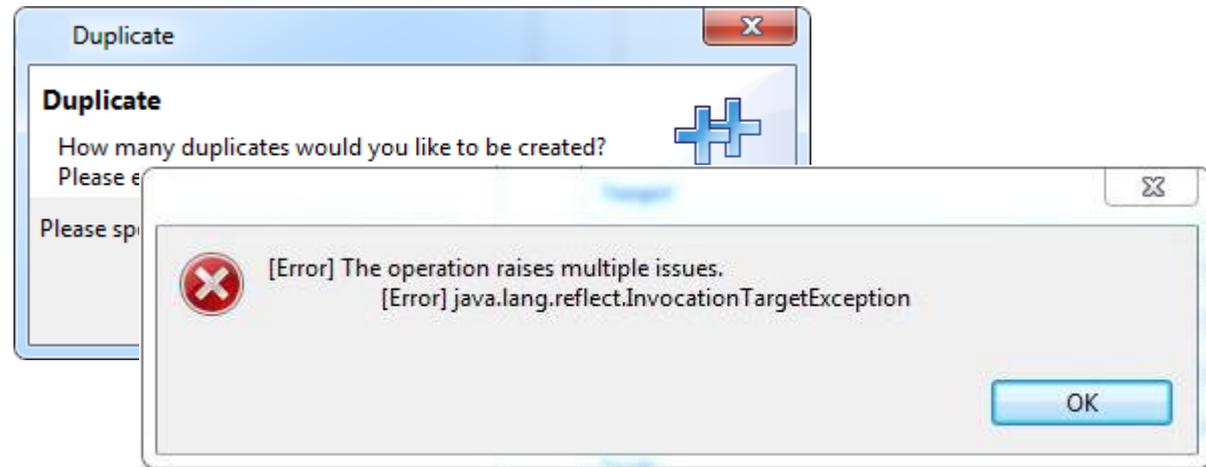


▶▶ The obvious solution...



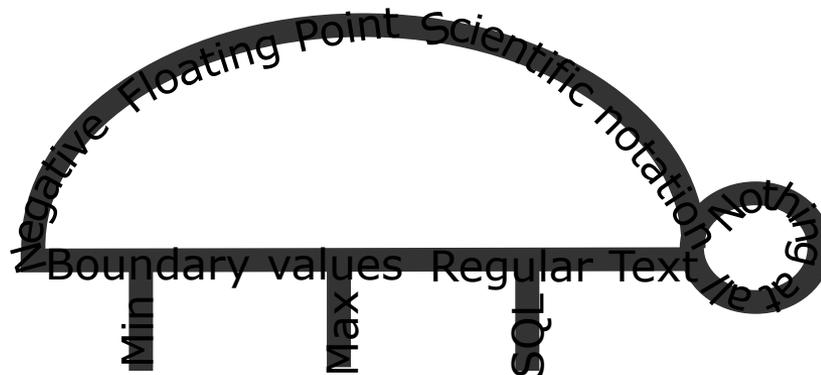
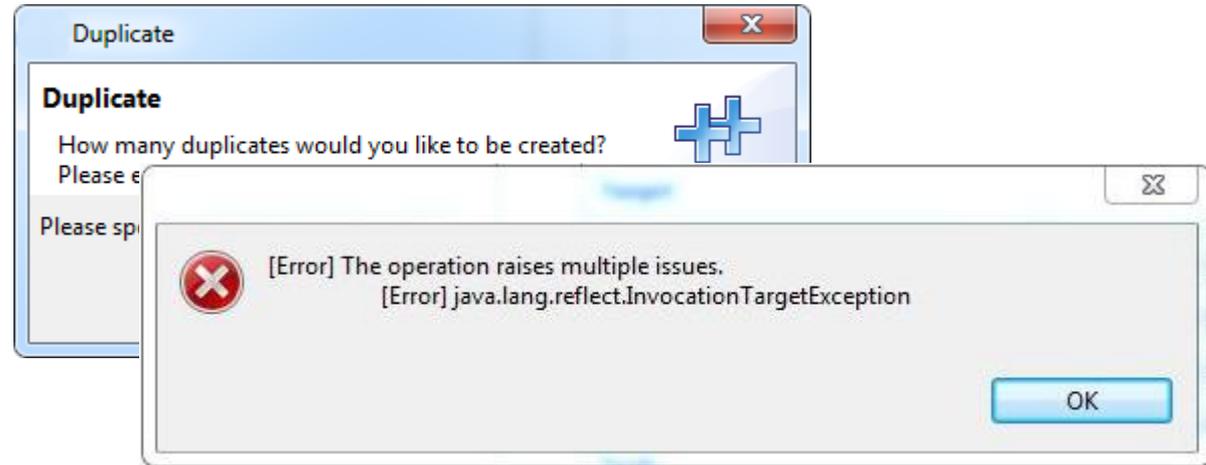
Preparation

Preparing for it by building muscle memory



Negative Floating Point Scientific notation Nothing
Boundary values Regular Text
Min Max SQL

Preparing for it by building muscle memory



Preparing for it by building muscle memory

This is a collection of testing challenges from varying sources.

Are you a good notetaker?

A step into a Brave New World

A test for India

A Tour through Europe

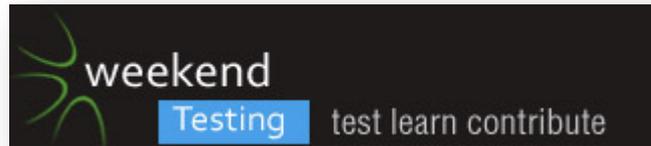
A Wiki in your Pocket

Battle of the Uri Shorteners

Beam Me Up

Blind as a Bat

<http://www.testing-challenges.org>



<http://weekendtesting.com>



<http://www.softwaretestingworldcup.com>

Coping

Coping by cheat sheets



Quality
Tree
Software

Test Heuristics Cheat Sheet *Data Type Attacks & Web Tests*

Data Type Attacks

Paths/Files **Long Name** (>255 chars) ▪ **Special Characters in Name** (space * ? / \ | < > , . () [] { } ; : ' " ! @ # \$ % ^ &) ▪ **Non-Existent** ▪ **Already Exists** ▪ **No Space** ▪ **Minimal Space** ▪ **Write-Protected** ▪ **Unavailable** ▪ **Locked** ▪ **On Remote Machine** ▪ **Corrupted**

Time and Date **Timeouts** ▪ **Time Difference between Machines** ▪ **Crossing Time Zones** ▪ **Leap Days** ▪ **Always Invalid Days (Feb 30, Sept 31)** ▪ **Feb 29 in Non-Leap Years** ▪ **Different Formats (June 5, 2001; 06/05/2001; 06/05/01; 06-05-01; 6/5/2001 12:34)** ▪ **Daylight Savings Changeover** ▪ **Reset Clock Backward or Forward**

Numbers **0** ▪ **32768** (2^{15}) ▪ **32769** ($2^{15} + 1$) ▪ **65536** (2^{16}) ▪ **65537** ($2^{16} + 1$) ▪ **2147483648** (2^{31}) ▪ **2147483649** ($2^{31} + 1$) ▪ **4294967296** (2^{32}) ▪ **4294967297** ($2^{32} + 1$) ▪ **Scientific Notation (1E-16)** ▪ **Negative** ▪ **Floating Point/Decimal (0.0001)** ▪ **With Commas (1,234,567)** ▪ **European Style (1.234.567,89)** ▪ **All the Above in Calculations**

Strings **Long** (255, 256, 257, 1000, 1024, 2000, 2048 or more characters) ▪ **Accented Chars** (àáâãäåçèéëìíîïðñòóôö, etc.) ▪ **Asian Chars** () ▪ **Common Delimiters and Special Characters** (" ' ` | / \ , ; : & < > ^ * ? Tab) ▪ **Leave Blank** ▪ **Single Space** ▪ **Multiple Spaces** ▪ **Leading Spaces** ▪ **End-of-Line Characters (^M)** ▪ **SQL Injection** ('select * from customer) ▪ **With All Actions** (Entering, Searching, Updating, etc.)

“Test heuristics cheat sheet” by Elisabeth Hendrickson, Dale Emery und James Lyndsay

<http://testobsessed.com/wp-content/uploads/2011/04/testheuristicscheatsheetv1.pdf>

Coping by cheat sheets

Web Application Security Testing Cheat Sheet

[hide]

1 DRAFT CHEAT SHEET - WORK IN PROGRESS

2 Introduction

3 Purpose

4 The Checklist

4.1 Information Gathering

4.2 Configuration Management

4.3 Secure Transmission

4.4 Authentication

4.5 Session Management

4.6 Authorization

4.7 Data Validation

4.8 Denial of Service

4.9 Business Logic

4.10 Cryptography

4.11 Risky Functionality - File Uploads

4.12 Risky Functionality - Card Payment

4.13 HTML 5

4.14 Error Handling

“Web Application Security Testing Cheat Sheet” from OWASP

https://www.owasp.org/index.php/Web_Application_Security_Testing_Cheat_Sheet

Coping by mnemonics

PACO

Portrait, **A**udio, **C**ommunication, **O**verlay

SFDIPOT (San Francisco Depot)

Structure, **F**unction, **D**ata, **I**ntegrations, **P**latform, **O**perations, **T**ime

CIDTESTD (Kid Tested)

Customers, **I**nformation, **D**eveloper relations, **T**eam, **E**quipment & Tools,
Schedule, **T**est Items, **D**eliverables

Coping by re-testing

The screenshot shows a software defect management tool interface. At the top, there is a navigation bar with tabs: Main, Desc., Analysis, Planning, Cat., Effort, Resol., Verf., Customer, Hist., and Changes. The 'Main' tab is selected.

Key fields in the 'Main' tab include:

- ID: DVTST00000060
- State: Submitted
- Summary State: not approved
- Linked
- Submit Date: 14.04.2015
- Keywords: (empty)

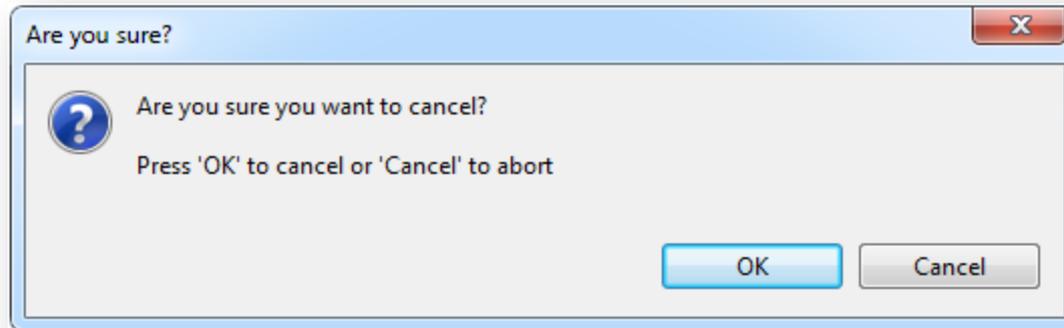
Below these fields, there are sections for 'Feature:' and 'Headline:'. The 'Headline' field contains the text: "NullPointerException if upper/lowercase of file names changes".

The 'Description:' section contains the following text:

Steps to reproduce:
- Import a file
- Change the upper/lowercasing of the file or directory on the haddisk
- Restart the import from the import history
=> NullPointerException, Import aborts

Exploiting it

Exploiting it by usability testing



Exploiting it by doing reviews

The screenshot shows a Microsoft Word document with the following content:

1 Document Management

Release			
Name	Role	Version	Date
Lieschen Müller	PMR		
	PLD		

Inspections			
Name	Role	Version	Date
Before release			
Peter Paul	PLD		
Andreas Ackermann	PL/Core Tool Team		
Jens Jossen	PMR		
After release			

Inspection Plan			
Name	When	Role	Comment
	[before release, after release, defined by document owner]		

Navigation pane (left sidebar):

- 1 Document Management
- 2 Project Overview
 - 2.1 Introduction
 - 2.2 Roles and Respons...
 - 2.3 Reference Docume...
- 3 Requirements Overview
- 4 Benchmark Definition
 - 4.1 Approach for defin...
 - 4.2 Use Cases
 - 4.2.1 Explore an EC...
 - 4.2.2 Trial-and-error
 - 4.2.3 Open-modify-...
 - 4.3 Benchmark projects
- 5 Input for performance r...
 - 5.1 Analysis of state-of...
 - 5.2 Performance goals ...
- 6 Performance Requireme...
- 7 References
 - 7.1 Glossary
 - 7.2 Abbreviations
 - 7.3 Requirements Source

But in the end...



...just try to get some sleep!

Your questions are welcome!

Please rate this talk at www.eclipsecon.org

Author:
Geyer, Tobias
Vector Informatik GmbH
@the_qa_guy