

# Automated Testing for Maximo using Google Selenium





## Profile

- Maximo for Oil and Gas/HSE lead QA  
(frcoelho@br.ibm.com)
- 8 Years testing Maximo, since versions 7.1.x
- 6 years automating tests for Maximo
- Selenium, Rational Functional Tester



## Why Automate?

- Prevents hours of exhaustive and repetitive manual efforts.
- It is a good practice to re-test your environment after an update/hot-fix is installed, in order to check that everything is still working. Specially customized applications.
- Suitable with whatever business process your company uses
- Create data using Maximo user interface.



## Difficulties to automate tests in Maximo 7.1 / 7.5

- Robust and unique User Interface.
- Object attributes are mostly not recognized by many automation tools.
- UI objects have dynamic IDs
- Now, automatable with Google Selenium



# Example of Object's attributes

Asset Management Interface Screenshot:

- Navigation: List View, **Asset**, Spare Parts, Safety, Meters, Specifications, Features, Relationships, Work, Work Zones
- Asset: ASSET1
- Status: NOT READY
- Linear?
- Site: BEDFORD
- Type: [Search]
- Asset Template:
- Attachments: [Icon]
- Moved?: [ ]
- Calibration?: [ ]

DOM Inspector (HTML):

```
<tr id="m73f28d7-co2_1_v1">  
  <td id="m73f28d7-co2_1" nowrap="nowrap" style="vertical-align:middle;" role="presentation">  
    <input id="m73f28d7-tb" class="fld text text fld_text" type="text" fldinfo="{\"saindex\":\"0\", \"length\":\"12\", \"afindex\":\"0\", \"eventpriority\":1, \"required\":true, \"inttype\":\"1\"}" work="1" ov="" value="" ae="setvalue" async="1" style="width:75px;\" maxlength="12" li="m73f28d7-img" ctype="textbox" role="textbox" aria-required="true" datatype="1" originalvalue="" prekeyvalue="ASSET1" title="ASSET1" keydown="false" changed_by_user="true" changed="true" stoptclick="true">  
    <img id="m73f28d7-img" class="dButton
```

\*Firefox browser with Firebug plugin



## What's new in Maximo 7.6?

- UI object attributes are the same
- **Object's IDs are now static (if not updated deliberately)**
- Even upgrading/installing new packages, IDs don't change
- Easily Automatable!



## About Selenium WebDriver

- WebDriver is designed to provide a simpler, more concise programming interface.
- Selenium-WebDriver makes direct calls to the browser using each browser's native support for automation
- Supports Java, JavaScript, PHP, Perl, Ruby, Python and C#.



## Demo

- Creating a new simple script. This example shows the most basic script you can do just to get started.
- How far you can get, depends on you.





## Oil and Gas/HSE projects numbers

- Oil and Gas/HSE 7.6.0.0
  - 225 test scripts created by 2 people.
- Oil and Gas/HSE 7.6.1.0
  - 430 test scripts created so far, by 3 people. All scripts created on 7.6.0.0 release will also be executed at the final phases of the project, without impacting on schedule, making sure that all existing functionalities still work.



# Thank you!

## Questions?





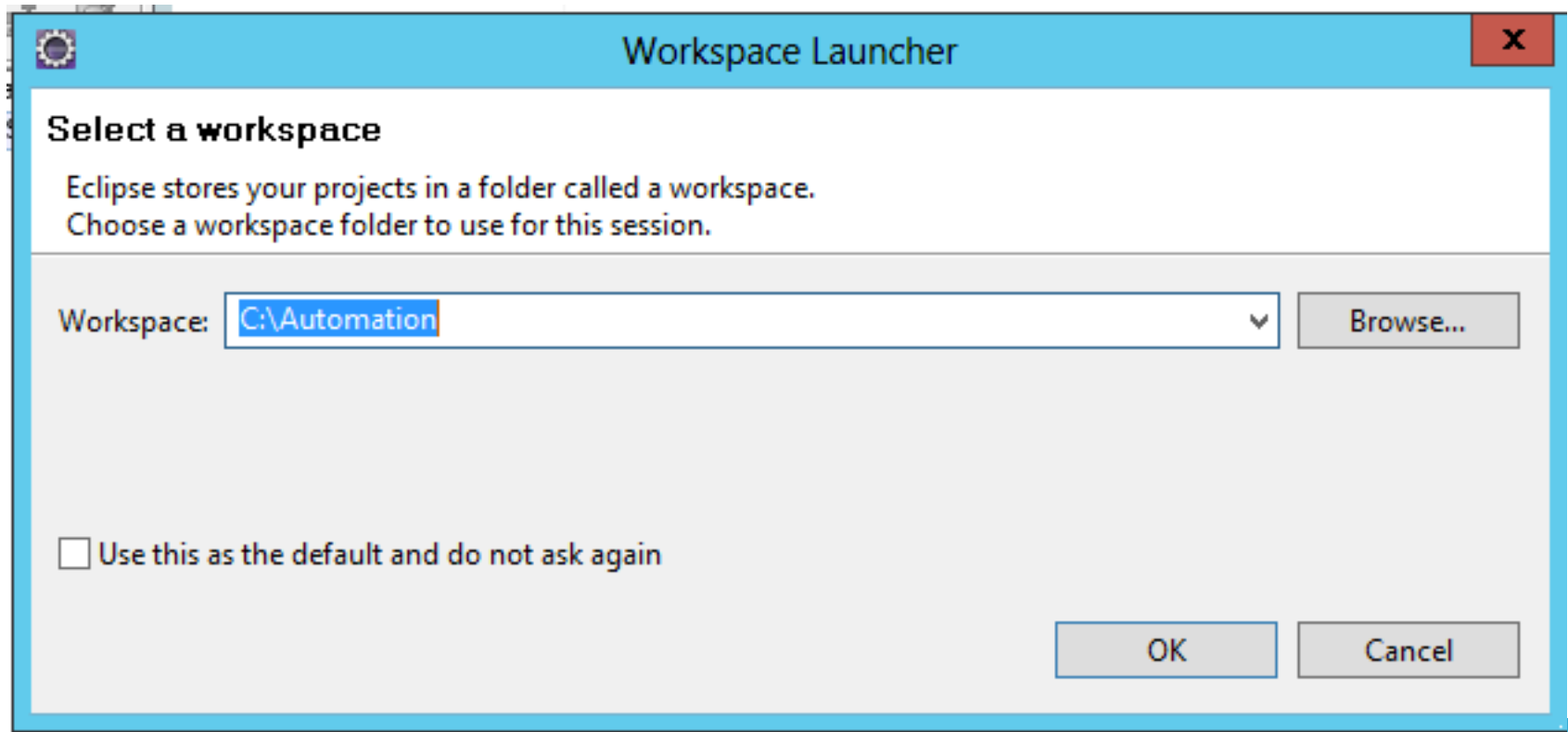
# FYI - Creating a new Eclipse workspace for Selenium

- Download selenium-server-standalone-2.53.1 from:  
<http://www.seleniumhq.org/download/>
- Download Chrome driver from:  
<https://sites.google.com/a/chromium.org/chromedriver/downloads>
- Create a new workspace for Eclipse, and place both files in it.  
(C:\Automation\SeleniumFiles)



## Creating a new Eclipse workspace for Selenium

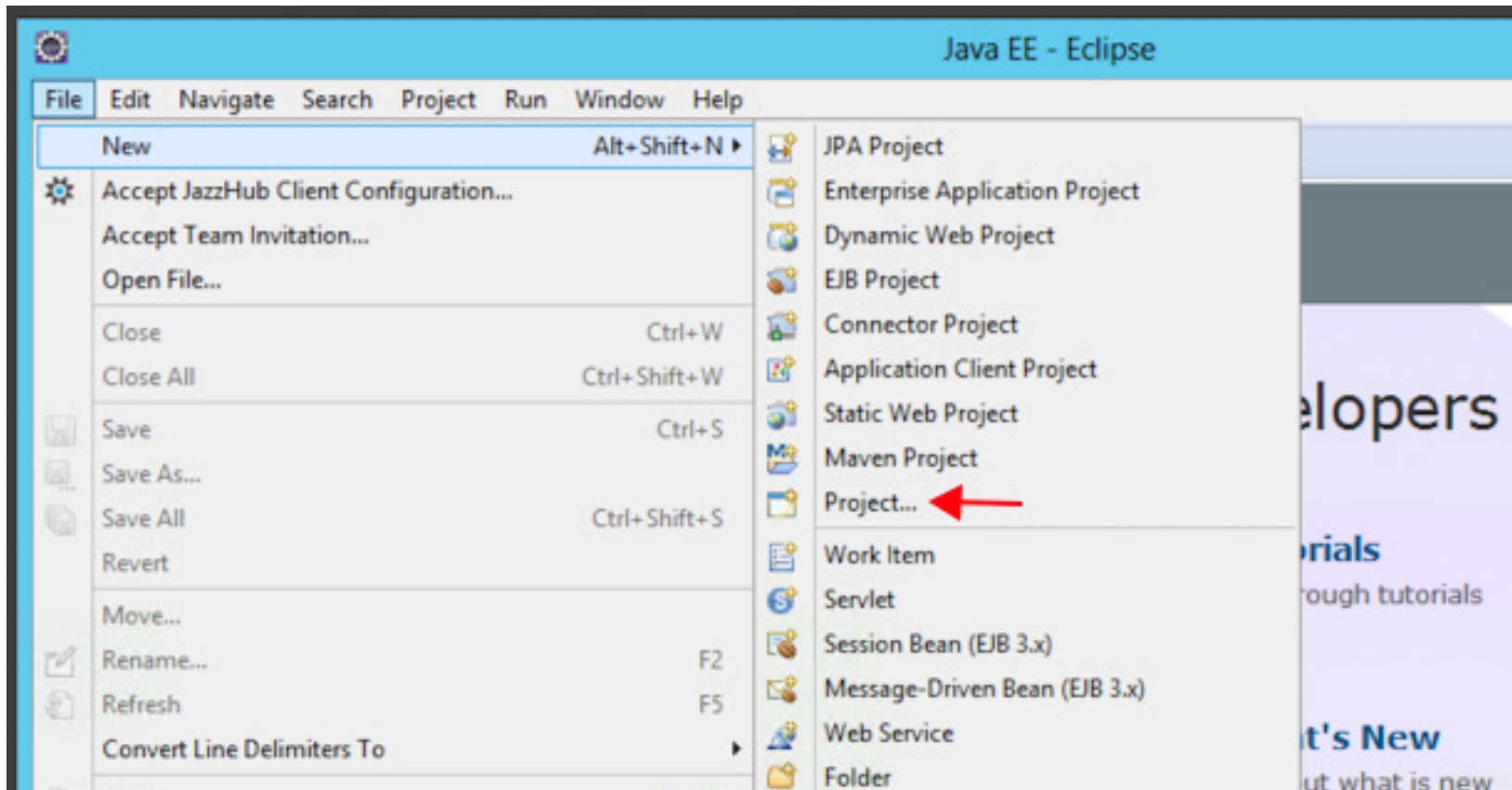
- Open an Eclipse instance, and select C:\Automation as your workspace location.





# Creating a new Eclipse workspace for Selenium

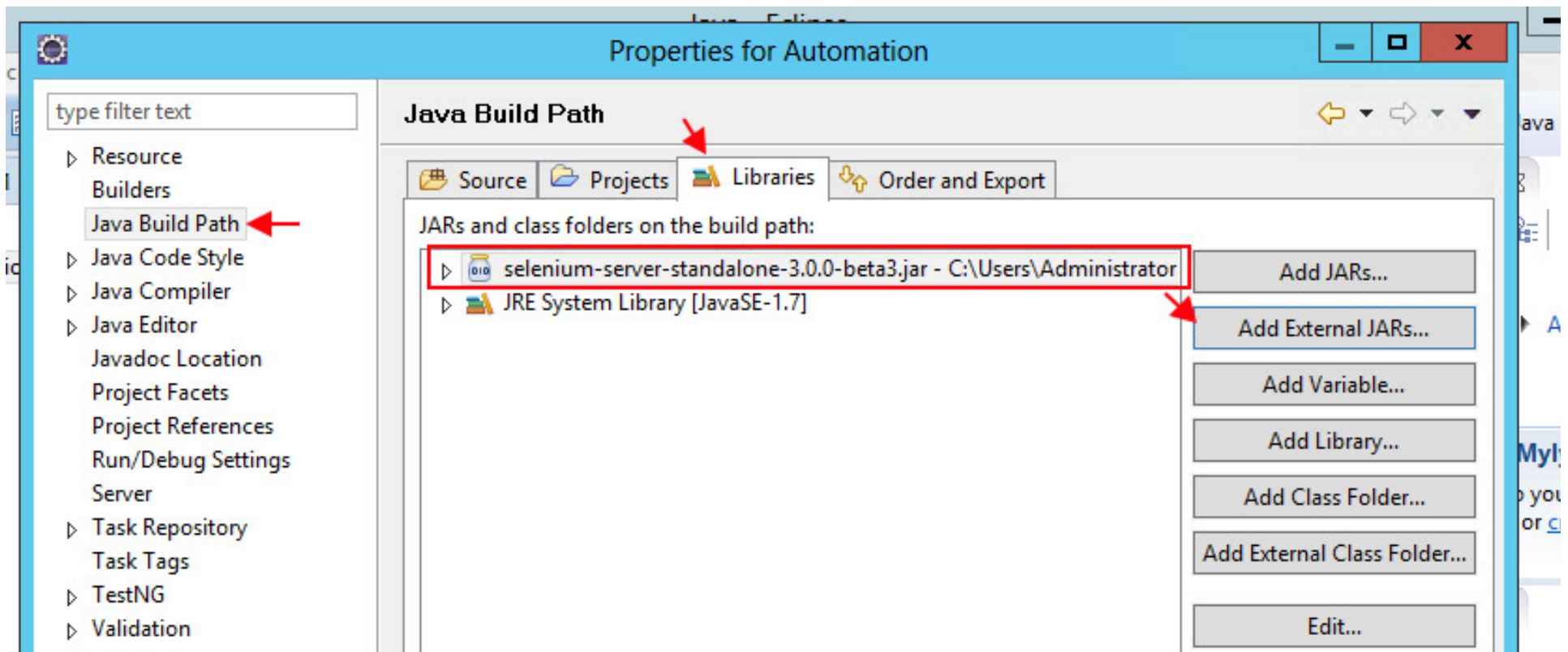
- Create a new Project, select Java Project, name it and click Finish:





# Creating a new Eclipse workspace for Selenium

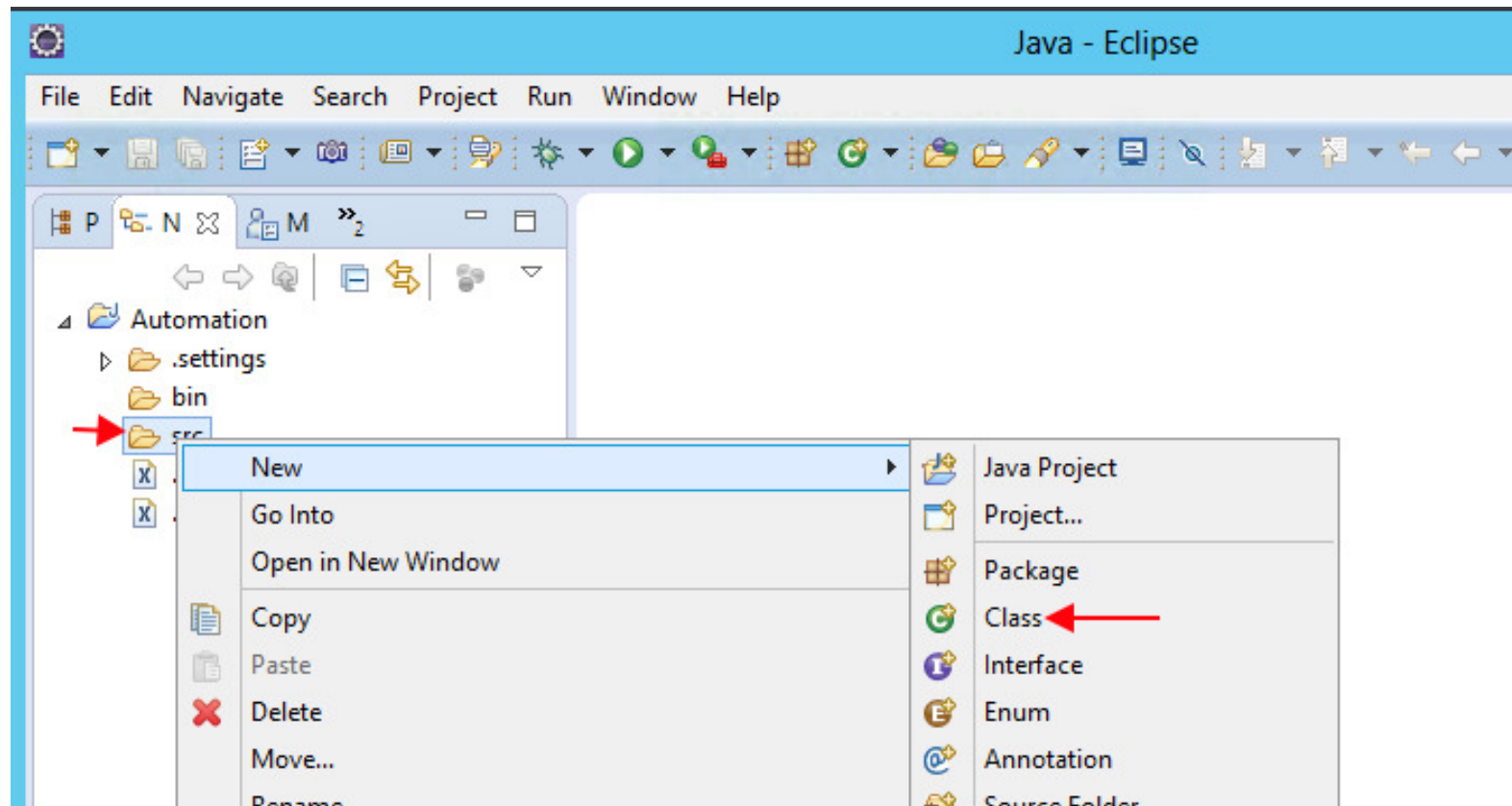
- Right click on Automation project and then click on Properties. Click on Java Build Path, then click on Libraries tab.





# Creating a new Eclipse workspace for Selenium

- Create a new Class, name it, and you should be good to start scripting.





# Creating a new Eclipse workspace for Selenium

- Copy and paste the code from the workshop below, and you can start testing on your own.
- For demonstration purposes, there are two different classes, one of them holds the scenario logics, and the other one holds the fields, buttons and icons mapping. The creation process is just the same.