Bending the Eclipse IDE to your will

EclipseCon Europe 2016
Target for this talk

How to extend or even replace stuff in the IDE
Creating a Plugin

New Plug-in Project

Templates
Select one of the available templates to generate a fully-functioning plug-in.

✔ Create a plug-in using one of the templates

Available Templates:
- Custom plug-in wizard
- Hello, World Command
- Plug-in with .target file generic editor extension
- Plug-in with multi-page editor
- Plug-in with popup menu
- Plug-in with property page
- Plug-in with view
- Plug-in with an Eclipse 4 handler
- Plug-in with an Eclipse 4 view
- Plug-in with an editor
- Plug-in with an incremental project builder
- Plug-in with sample help content

This template creates an Eclipse 4 plugin with a model application fragment containing:
- a 'Hello world' command
- a 'Hello world' handler
- a menu contribution to the main menu
- a menu contribution to any popup

It uses the standard xpath:/: notation to add this view in any application model.

The generated plugin is also compliant with an Eclipse 3 application running with the compatibility layer.

The handler can be triggered by selecting **E4 Menu > Hello world** in the main menu or in any view popup menu.

Extensions Used
- org.eclipse.e4.workbench.model
Wizards for almost everything

- Add a new wizard
- Add a new menu
- Contribute to an existing menu
- Create a new view
- Create a new editor
- and much more...
Testing a plug-in or a change

- Right-click on your plug-in -> Run As -> Eclipse Application
Deploying a plug-in

- File -> Export -> Deployment plug-ins and fragments
- Directory -> Place into dropins folder
Find Entry Points of the IDE
Eclipse Spys

- PDE Plugin Spy (ALT + SHIFT + F1)
- PDE Menu Spy (ALT + SHIFT + F2)
- E4 Model Spy (ALT + SHIFT + F9)
Files and Source Code

- ResourcesPlugin and IWorkspaceRoot
  - org.eclipse.core.runtime
  - org.eclipse.core.resources

- JDT Abstract Syntax Tree
  - org.eclipse.jdt.core
Quick Assist and Quick Fixes

• Quick Assist
  • Can be applied to general source code

• Quick Fixes
  • Find ProblemIds for Quick fixes
  • Can be applied for markers (errors, …)
Changing the IDE directly

```
git clone -b master --recursive
git://git.eclipse.org/gitroot/platform/eclipse.platform.releng.aggregator.git

Do your changes...
```

```
mvn clean verify
```
Why should I care?

- Automate manual efforts for your needs
- Put several steps together
- Be more productive!
Further Reading...

- [https://github.com/vogellacompany/codemodify](https://github.com/vogellacompany/codemodify) Example for code refactoring JUnit 3 tests to JUnit4
Questions?
Thank you

For further questions:

http://www.vogella.com

Need commercial support:
sales@vogella.com
Evaluate the Sessions
Sign in and vote at eclipsecon.org

- 1 0 + 1