

# An Epic Split between OSGi Tooling Eclipse PDE and Bndtools



created by [Peter Kirschner](#) for [EclipseCon Europe 2014](#)

Seminarräume 1 - 3 - Wednesday, October 29, 2014 - 10:30 to 11:05

copyright for Epic Split Theme is with Volvo Trucks

[printable version](#)

# About Me



- name: Peter Kirschner
- profession: IT software engineer
- email: peter@kirschners.de
- twitter: [@peterkir](#)
- github: <https://github.com/peterkir/ece2014>

# Table of contents

- Motivation
- Comparison PDE to Bndtools
- Development Setup & Update
- Programming
- Conclusion

# Motivation

- PDE pain points
- Why bnd/Bndtools

# PDE pain points

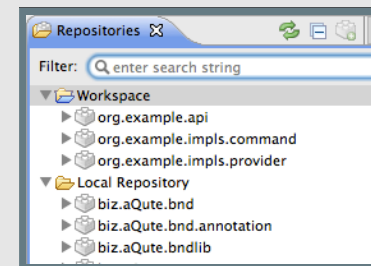
- dependency management is difficult
- target horror
  - configuration complex
  - reloading is not fail safe
- launch configurations
  - what the hell has PDE loaded in the runtime
  - where do these runtime settings come from

# Why Bndtools

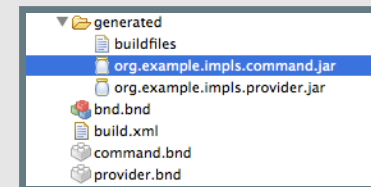
automated import package analysis  
including versioning



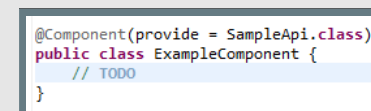
powerful OSGi bundle repository  
management



instant Jar bundle creation



annotation based Declarative Service  
implementation



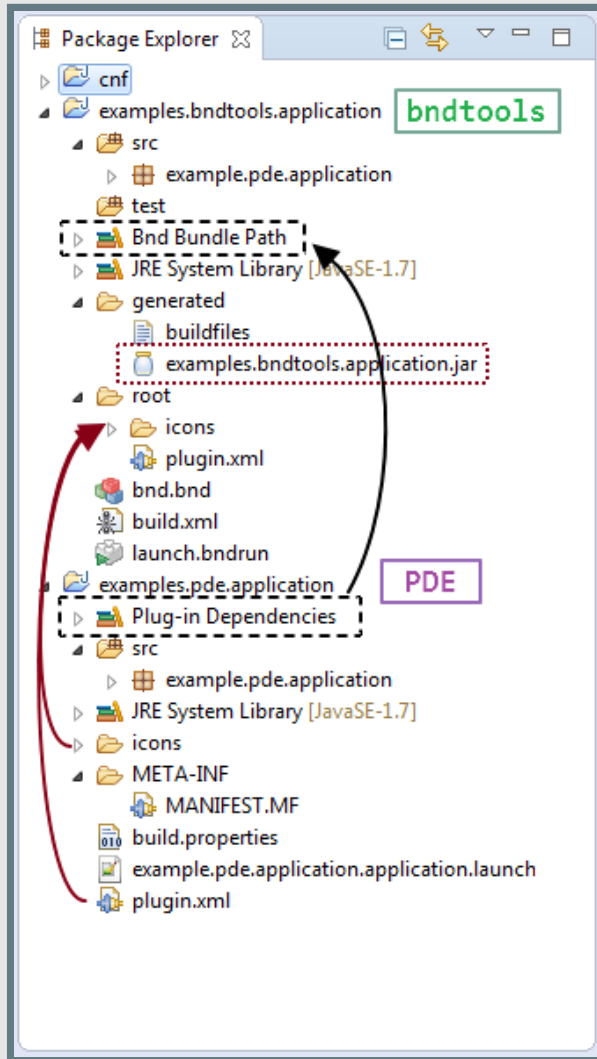
# Comparison of PDE and Bndtools

- Project Layouts
- IDE Bundle Handling

# Project Layouts

single/multiple bundles  
from one project

build path on  
eclipse project level





# IDE Bundle Handling

- PDE
  - creates "virtual bundles" for execution Launches
  - PDE build/export required to create concrete jar files
- Bndtools
  - creates physical jar file inside generated folder
  - launches execute this jar file inside runtime

# Concurrent areas - Mixed-Mode

- Project Build Path
- Project Interdependencies
  - PDE depends on BND
  - BND depends on PDE
- Compiler Settings

# Project Build Path

- PDE
  - projects share a common target
  - high complexity selecting import-packages
- Bndtools
  - configured for each project
  - reduces number of import-package
- **bnd projects have a precise import-package scope**

# PDE depends on BND

- OSGi dev with PDE target
  - BND creates generated/bundle.jar
  - p2 publisher generates p2 repo from bnd jar files
  - include p2 repo in target and reload it

# BND depends on PDE

- BND project depending on PDE bundle
  - PDE feature export/build creating p2 repo
  - obr repo indexing
  - reference and refresh cnf
- **Luckily we don't required this**

# Synchronize Compile Settings

## PDE compilation on target

```
<target_prj>/workspace.target  
<pde_prj>/classpath  
<pde_prj>/settings/org.eclipse.jdt.core.prefs  
<pde_prj>/META-INF/MANIFEST.MF -> Bundle-RequiredExecutionEnvironment  
<pde_prj>/build.properties -> jre.compilation.profile
```

## Bndtools

```
<cnf_prj>/ext/repositories.bnd  
<bnd_prj>/classpath  
<bnd_prj>/settings/org.eclipse.jdt.core.prefs  
<bnd_prj>/bnd.bnd
```

## Synchronize PDE and bnd compilation via import of JDT settings

# Development setup

- Initial Dev Setup
- Updating Setup or Target
- Mixed-Mode Helpers

# Initial Dev Setup

- Eclipse SDK to create/update -> **bootstrap IDE**
  - install **IDEfix** (eclipse + Bndtools + ...)
  - mirror required repos locally
- **IDEfix** with eval workspace
  - adapt Bndtools cnf
- **IDEfix** product workspace
  - use custom cnf and develop product



# Update Dev Setup

- **bootstrap IDE**
  - mirror new bndversion and install new IDEfix
  - mirror and index new target repositories
- launch **IDEfix** with an new eval workspace
  - eval bnd updates on cnf
- launch **IDEfix** with new product workspace
  - compare and configure cnf
  - import your projects

# Mixed-Mode Helpers

- p2 repository mirroring with OBR indexing
- IDE bootstrap scripts mirror, install, launch templates
- bnd2pde -> p2 publishing and PDE target creation
- [PDE TargetReloader Plug-in](#)

# Development

- Mixed-Mode
- Coding

# Mixed-Mode

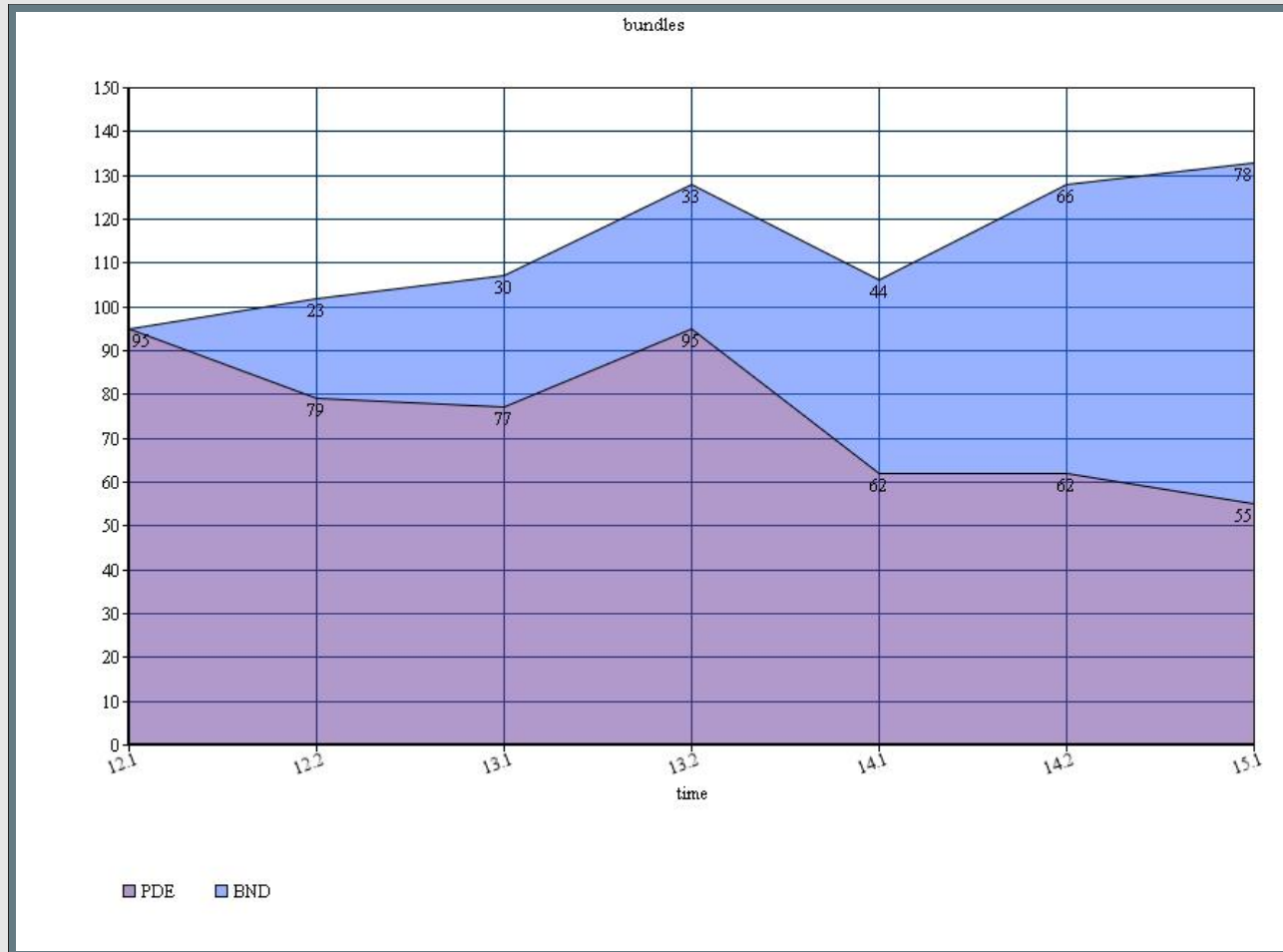
- After modifications on bnd bundles
  1. bnd2pde build
  2. target reload required
  3. validation via view "TargetPlatform State"
- Debugging source code editor is showing read-only bnd target bundles
  - editing impossible (uncommon behaviour)
  - open the appropriate source type with Ctrl+Shift+T and CamelCase notation

**Coding**  
**with Bndtools can leverage**  
**the Joy of OSGi**

# Conclusion

- Our Migration Progress
- Migration Digest

# Our Migration Progress



# Migration Digest

- Mixed-Mode is hard -> try to migrate fast
- Defaults are not ideal for Enterprise Environments
- Bndtools Windows Committers wanted - Windows is special
- Work is more comprehensible
- Benefits for us exceeded the tooling problems

**Migration is feasible, but is not for free!**



???

- contact [me](#) for more information
- example source on [github](#)
- example project is work in progress

# Credits for

- my suffering colleagues for their
  - valuable feedback
  - and patience in adopting new technologies
- Peter Kriens, Neil Bartlett, BJ Hargrave
- and all other bnd/Bndtools/bindex commiter and contributors
- [REVEAL.JS](#) - HTML presentations by Hakim El Hattab

# Volvo Trucks - Making of ...

VAN DAMME - Real split between two trucks (HD) - Complete story



[short version \(1:16mn\) - commercial spot only](#)

# BND and Bndtools resources

- [BND](#)
- [Usergroup](#)
- [Source on Github](#)
- [Continuous Builds from Cloudbees website](#)
- [bug reports/source code on GitHub](#)
- [continuous build on cloudbees](#)
- [usergroup in GoogleGroups](#)

# standard references

- OSGi specifications
- LDAP search filter syntax
- Semantic Versioning 2.0.0