FROM NOTHING TO COMPLETE ENVIRONMENT
WITH MAVEN, OOMPH & DOCKER

Max Bureck, 21. June 2017
FROM NOTHING TO COMPLETE ENVIRONMENT

1. Disclaimer

2. Motivation

3. Demo

4. Recap, Conclusion, and Future Possibilities
FROM NOTHING TO COMPLETE ENVIRONMENT

http://memegenerator.net/instance/78175637
Disclaimer: Prerequisites are Maven, Oomph & Docker 😊

- Maven >= 3.0
- Latest Eclipse installer
- Docker
Windows 10: Easiest Way to Make Docker Daemon Accessible From Eclipse

FROM NOTHING TO COMPLETE ENVIRONMENT
FROM NOTHING TO COMPLETE ENVIRONMENT

Motivation: Target Domain Introduction

Writing a transformation service
FROM NOTHING TO COMPLETE ENVIRONMENT

Motivation: Target Domain Introduction

- Developing ModelBus services should be simple
- Usually first task for new team member:
  - Read ModelBus user guide
  - Create example ModelBus services
Motivation: Setup Experience, Step 1

- Setup Action:
  - Install ModelBus server
  - Configure ModelBus server

- Problems:
  - User forgets to set mandatory environment variable
  - User forgets to add SVN plugins (for legal reasons not part of ModelBus distro)
Motivation: Setup Experience, Step 2

- Setup Actions:
  - Install Eclipse
  - Install ModelBus TeamProvider plugins
  - Configure TeamProvider and target platform

- Problem:
  - User does not add plugins for standalone services to target platform
Motivation: Setup Experience, Step 3

- Setup Action:
  - Create projects for service interface, provider - and consumer service

- Problem:
  - User forgets to add mandatory plugin dependency
Motivation: Setup Experience, Step 4

- Setup Action:
  - User creates launch configurations for provider and consumer projects

- Problem:
  - User forgets to set one or more start-levels for plugins
Motivation: Setup Experience, Step 5

- Setup Action:
  - User creates product definition for services (for standalone executable)

- Problems:
  - User forgets to set one or more start-levels for plugins (again)
  - Forgotten plugins in product definition
FROM NOTHING TO COMPLETE ENVIRONMENT

Motivation: Setup Experience, Step 6

- Bonus Action:
  - Create Maven Tycho build scripts

- Problems:
  - Bundle IDs / Feature IDs have to match POM artifactIds
FROM NOTHING TO COMPLETE ENVIRONMENT

Motivation: Setup Experience Conclusion

SIMPLE?
YOU WISH!

https://imgflip.com/i/1qsphq
Motivation: Improving Setup Experience

- Use automation to get rid of error prone manual tasks of
  - Installation of ModelBus server
  - Installation and configuration of Eclipse + plugins
  - Creation of service interface, provider, and consumer plugins
  - Creation of launch configs and product definition
  - Creation of Maven Tycho build scripts
Recap, Part 1

- We used Maven archetype to generate
  - Projects (service interface, provider, consumer, features, products)
  - Maven build scripts
  - Oomph setup file
  - Dockerfiles for ModelBus server, provider service, consumer service
  - Launch configs for build, run, debug
Recap, Part 2

- We used generated Oomph setup to
  - Install Eclipse
  - Install plugins into Eclipse
  - Configure Eclipse + plugins
  - Configure target platform
Conclusion

- If your platform is based on OSGi / Eclipse tech: Consider Maven archetypes
- Use Oomph for (reproducible) Eclipse setups
- Consider Docker if your Eclipse based platform integrates with network services
Going Even Further (When Possible)

- Install prerequisites via package managers (apt, chocolatey, homebrew, etc.)
  - Single shell script call for setup, literally starting from nothing
- Starting eclipse installer with Oomph setup file from command line
- Generating CI configuration (Jenkinsfile, .travis.yml, .gitlab-ci.yml, etc.)
FROM NOTHING TO COMPLETE ENVIRONMENT

Please Vote and Leave Comments

Evaluate the Sessions

Sign in and vote at eclipsecon.org

-1 0 +1
Thanks!
Questions?
CONTACT

Fraunhofer FOKUS
Kaiserdin-Augusta-Allee 31
10589 Berlin, Germany
www.fokus.fraunhofer.de

Max Bureck
Senior Researcher
max.bureck@fokus.fraunhofer.de
Tel. +49 (0)30 3463-7321