Tycho and CBI adoption

Feedback from the trenches

EclipseCon Europe 2012

About the speakers

Krzysztof:

- Full Name: Krzysztof Daniel
- Twitter: @kda
- Eclipse packager in Fedora Linux
- Contributor to Platform Tycho-based build
- Tycho sceptic

Mickael:

- Full Name: Mickael Istria
- Twitter: @mickaelistria
- JBoss Tools & JBoss Developer Studio build
- Committer on SWTBot, Nebula, GMF-Tooling (mainly build stuff)
- Tycho enthusiast

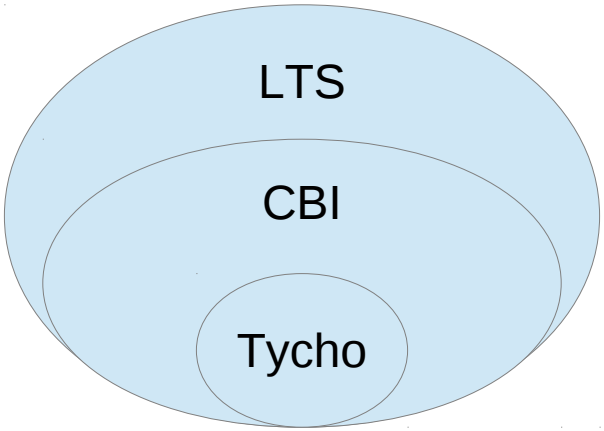
Agenda

1. CBI/Tycho build - clarification
2. PDE Build vs Tycho: Differences
3. Short migration guide (with tricks)
4. Case study: SWTBot
5. Case study: Eclipse Platform
6. Questions and answers

A small Clarification

CBI is an initiative to lower the contribution barrier.

Tycho is a technical solution.



redhat.

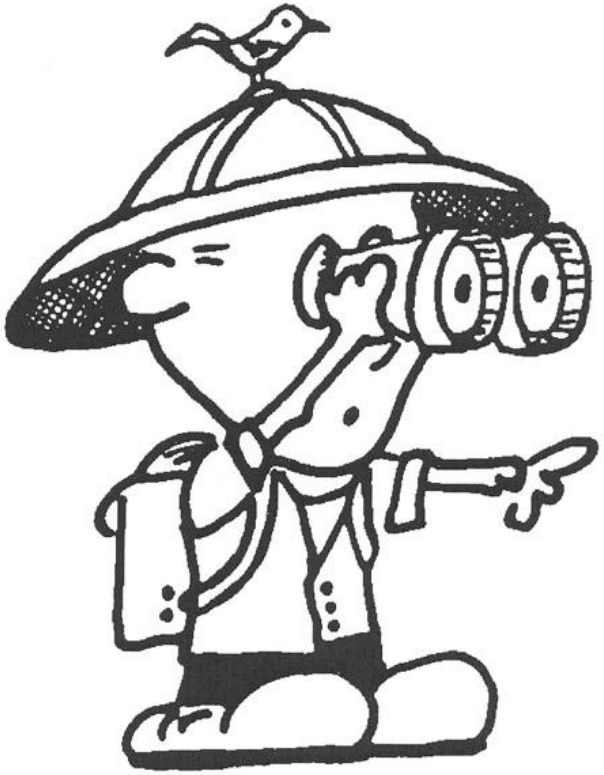
Chulk and Cheese

PDEBuild is...	Tycho is...
Ant-based	Maven-based
Driven by scripts	Driven by lifecycle and descriptors (pom)
totally open to hacks Target-platform aware	strict & constraining Target-platform aware (p2 only)
customizable via "Callbacks"	customizable via plugins

Chulk and Cheese (continued)

PDEBuild...	Tycho...
1 script = N bundles/features	1 pom = 1 artifact
is a monolithic build	Allows incremental build (artifact after artifact)

Knowing your enemy: build.properties



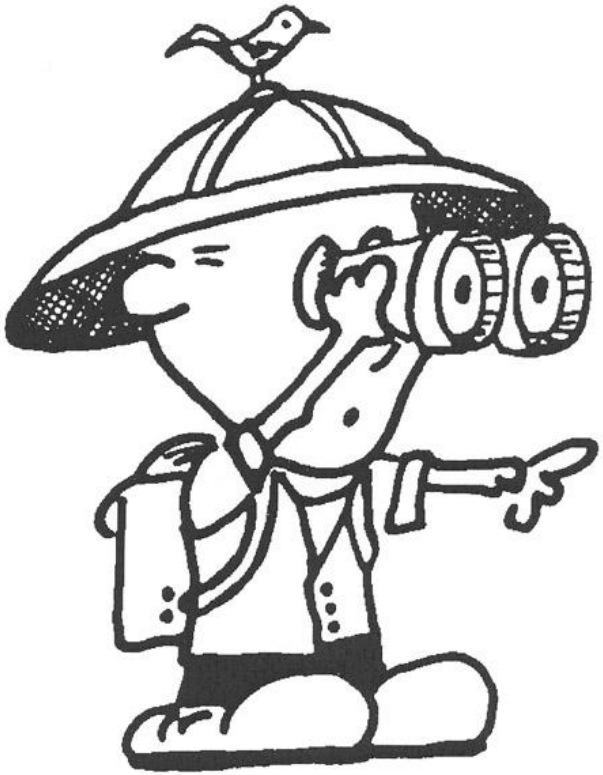
custom=true

customBuildCallbacks=...

```
$ find . -name build.properties \
                    -exec grep -H
                custom  {} \;
```

redhat.

Knowing your enemy: repository shape



nested artifacts:

*.dll, *.class, *.jar, *.jnilib, *.so

content in features

Documentation generation

cycles

redhat.

Migration Guide: 101

Create a pom.xml for each artifact

```
mvn org.eclipse.tycho:tycho-pomgenerator-
plugin:0.16.0:generate-poms
-DgroupId=<MY.PROJECT>
```

http://wiki.eclipse.org/Tycho/Reference_Card#Generating_POM_files

Migration Guide:
A partial Tycho build

While keeping PDE/Build for production...

1. Generate poms
2. Convert problematic artifacts into p2 repo
3. Use the pre-compiled artifacts in your build as required
4. Fix one bundle at a time.

Migration Guide: be modular !

Any module (plugin, feature...) should be able to build on its own (and only on it's own).

mvn verify should work from anywhere in your source tree -as long as dependencies can be resolved-.

Necessary for incremental builds (successive *mvn install*)

Migration Guide - don't be afraid to...

Refactor: Maven will force you to refactor, but don't be afraid, in the end everything looks cleaner.

Example: GMF Tooling had to move documentation to a dedicated plugin.

Iterate: Start with the ugliest solution and improve later. Ugliest Maven solution is already an improvement over PDE/Build.

Migration Guide - customize

PDE/Build flow is replaced by **Maven lifecycle (phases)**

PDE/Build customCallbacks are replaced by usage of **Maven plugins**

A Trap: Implicit dependencies between plugins

Example: javadoc generation

```
-sourcepath "
;../org.eclipse.ant.core/src ...
-d reference/api -classpath @rt@ ;../com.ibm.icu_*.jar ...
```

Tycho solutions:

- Explicit your dependencies in your bundle
- Explicit dependencies when invoking a Maven plugin
- Refactor

to the Maven world and its **plugin ecosystem**

redhat.

Migration Guide - iteration #1

Use **maven-antrunner-plugin** to invoke your customCallbacks/build.xmls.

If customCallbacks require some Eclipse-specific Ant tasks, use **tycho-eclipserun-plugin** with org.eclipse.ant.core.antRunner application

Migration Guide – native libraries

- use maven-antrunner-plugin
- remember to propagate errors and fail the build when necessary
- use any build phase before the "compile" - (recommended "generate-resources")
- Look forward to maven-nar-plugin
- Try to "flatten" artifacts

Migration Guide - help generation

use tycho-eclipserun-plugin

adjust all the paths or copy required bundles

Migration Guide - innern jars

tycho-custom-bundle-plugin

- creates inner jars from fileset
- used in JDT
  - batch-compiler
  - antadapter

Migration Guide - cool plugins (Maven)

maven-dependency-plugin:get and **maven-download-plugin:wget** mojos to include other stuff in your jar.

Still missing: a real way to do filesystem operation with Maven (always use Ant plugin…)

Migration Guide - cool plugins (Tycho)

- tycho-buildtimestamp-jgit
  - version changes after code change
  - qualifier-commit consistency

- eclipse-jarsigner-plugin

Migration guide - sources

tycho-source-plugin generates a source bundle made of sources for your bundle.

tycho-source-feature-plugin generates a feature made of all <referencedBundles>.source

Very good in most cases, but don't allow big customization for tricky cases

Migration Guide: Tests made "native"

eclipse-test-plugin and **tycho-surefire-plugin** run test in the minimal necessary environment (as specified in MANIFEST.MF)

Migration Guide - very custom stuff

Create your own Maven plugin for that.

Try to get it generic enough to share this with the community as part of Tycho.

Truly liberating build

- Offline
  - only once your Maven repo contains the right stuff
  - using "offline" target platforms and repo (file://...)
- Forever reproducible (p2 repo sustainability?)
  - Need to keep your target platform
- Automated
  - no single item should be created outside of maven.
- Default

Case study: SWTBot

- *Vanilla* Tycho use-case
- No workaround
- Just *mvn verify* and you get a ready-to-publish p2 repo
- Simply added code coverage and Sonar support by using Maven plugins

Case Study: Platform

- it is not default **at all**
- "natives" repo
- failing Eclipse build on MacOS (384482)
- native parts not built (except in Fedora)
- dependency trap workarounded by:
  - adjusting paths
  - copying deps (for help)

Q&A