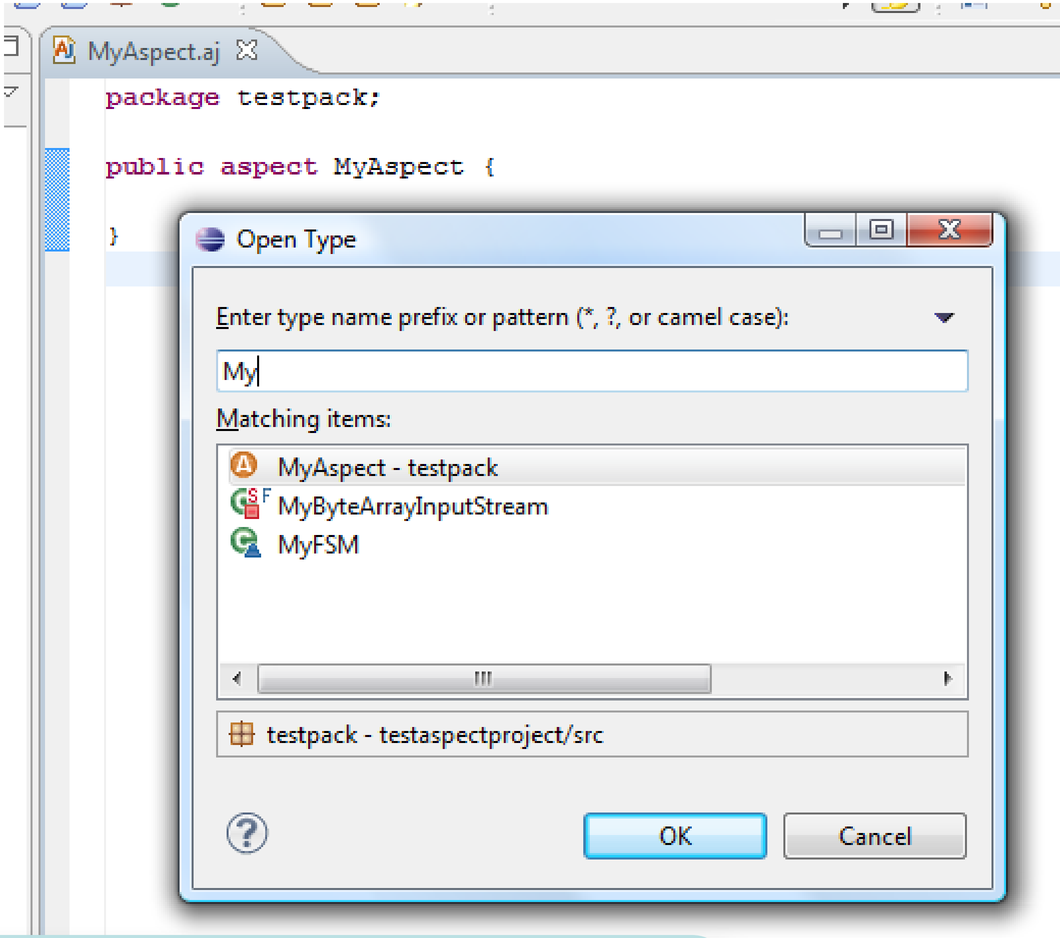
What's New in Equinox Aspects

Martin Lippert (akquinet it-agile GmbH)

| April 21st, 2009

Adoption

- **AJDT** now uses Equinox Aspects to weave into JDT
- **Scala IDE for Eclipse** uses the same JDT weaving



```
package testpack;

public aspect MyAspect {

}
```

Enter type name prefix or pattern (*, ?, or camel case):

My

Matching items:

- MyAspect - testpack
- MyByteArrayInputStream
- MyFSM

testpack - testaspectproject/src

Short Talk by Andrew Eisenberg at EclipseCon 2009: **Aspects Everywhere: Using Equinox Aspects to Provide Language Developers with Deep Eclipse Integration**

Weaving Dynamics

- "Be a good citizen of the OSGi community"
  - Weaving when resolved

- You can install and uninstall aspects at runtime
  - Refreshing other bundles
  - But: other bundles need to be dynamic-aware

Weaving Control

- **Major changes in M7 !!!**
  - You get much more control

- Manifest-only declaration of aspects
- Aspect-Policy for exported aspects
- Apply-Policy for imported aspects

Manifest-Only Declaration

- You don’t need an aop.xml file anymore
- Declare your aspects within the manifest

```
Export-Package:
 mypackage;aspects="FooAspect,BarAspect"
```

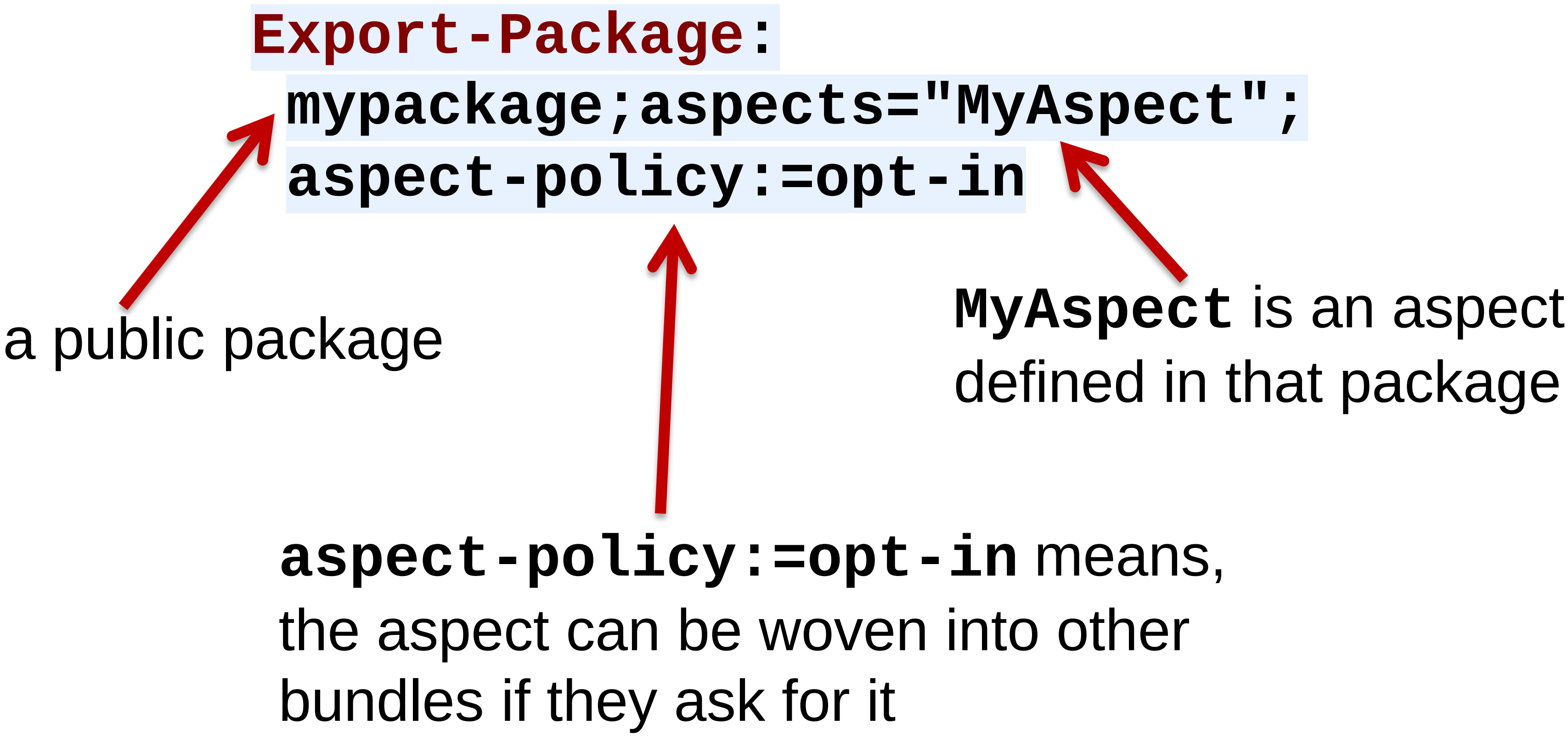
Tells the runtime that **`mypackage`** is visible to other bundles and contains two aspects for weaving

Aspects-Policies

- The **Aspect-Policy** is defined by the aspect bundle for the exported aspects

- Two variants available:
  - **Opt-In:** Importing bundles need to explicitly ask for them
  - **Opt-Out:** Importing bundles get the aspect automatically unless they prohibit weaving for it (default)

Aspect-Policy Example



```
Export-Package:
 mypackage;aspects="MyAspect";
 aspect-policy:=opt-in
```

a public package

`MyAspect` is an aspect defined in that package

`aspect-policy:=opt-in` means, the aspect can be woven into other bundles if they ask for it

Apply-Policy

- When you import an aspect, you can explicitly tell the system whether to apply the aspects or not

- Two variants:
  - **apply-aspects:=true:** Imported aspects are woven into the importing bundle (regardless of the aspect-policy, default)
  - **apply-aspects:=false:** Imported aspects are **NOT** woven into the importing bundle (regardless of the aspect-policy)

Apply-Policy Example

```
Import-Package:
 mypackage;apply-aspects:=true
```

Import the package

`apply-aspects:=true` means to weave aspects that are imported via the package to weave into the bundle

Caching

- The goal: **Zero overhead for cached scenario**
  - **This is achieved with 1.0 M7**

- Many tweaks already done
  - Fast cache read/write/lookup IO
  - Just one load op (not the default and then the cached)
  - No cache lookup for non-woven bundles
  - Awareness of bundle versions + updates
- Latest addition:
  - Caching for generated classes (around closures)

Asynchronous Cache Writing

- Benefits:
  - Better performance for loading thread (doesn’t have to wait for cache IO)
  - Less concurrency for IO itself (simpler implementation)
  - More robustness for the system in case of IO problems

- Implementation:
  - Concurrent bounded queue (even if cache writing hangs, memory usage is bounded)

Performance Improvements

- Improved AspectJ load-time weaving
- Many tweaks for caching
  - make the cached scenario really fast
- Lazy creation of weavers
  - create them only if they are really needed
- Less XML parsing
  - aop.xml files are parsed only once

P2 Installation

- P2 metadata added
- Installable via P2
  - Including automatic changes to config.ini
- AJDT already provides an update site
  - http://download.eclipse.org/tools/ajdt/35/dev/update
  - But weaving service is not started automatically

- Still some issues:
  - Non-writable locations
  - Different locations
  - Manual configuration cumbersome

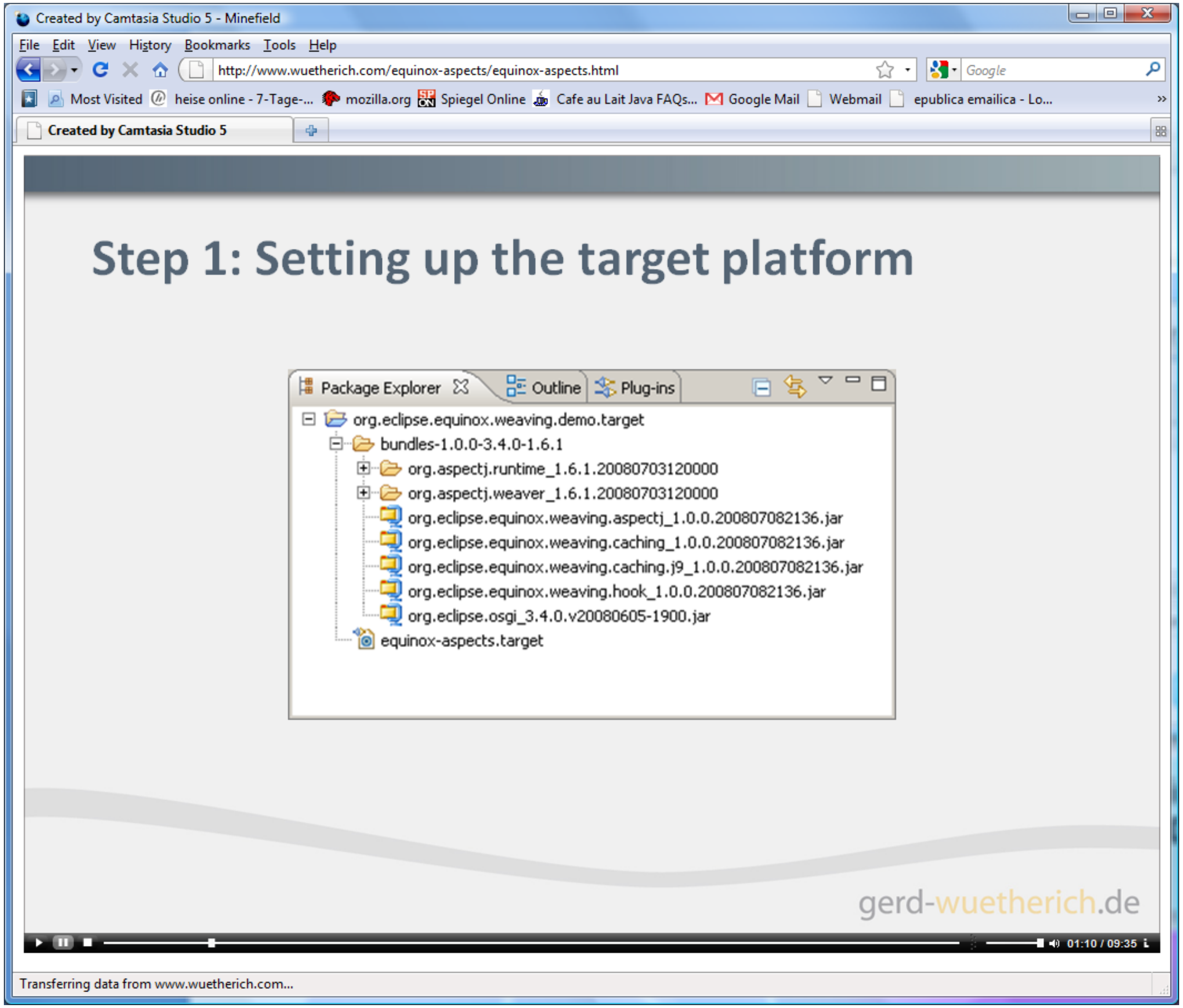
Other Weavers

- The base hook provides weaving API
  - And calls the weaver at classloading time
- The weaving itself is provided as OSGi service by other bundles
  - Pluggable weavers
  - Currently: AspectJ weaving
- No need to re-implement caching
- *This is not dynamic at the moment*

Screencast

- Introductory material

- Contributed by Gerd Wütherich (Thanks, Gerd!)


Created by Camtasia Studio 5 - Minefield
File Edit View History Bookmarks Tools Help
http://www.wuetherich.com/equinox-aspects/equinox-aspects.html
Google
Most Visited
heise online - 7-Tage-...
mozilla.org
Spiegel Online
Cafe au Lait Java FAQs...
Google Mail
Webmail
epublica emailica - Lo...
Created by Camtasia Studio 5
Package Explorer
Outline
Plug-ins
gerd-wuetherich.de
01:10 / 09:35

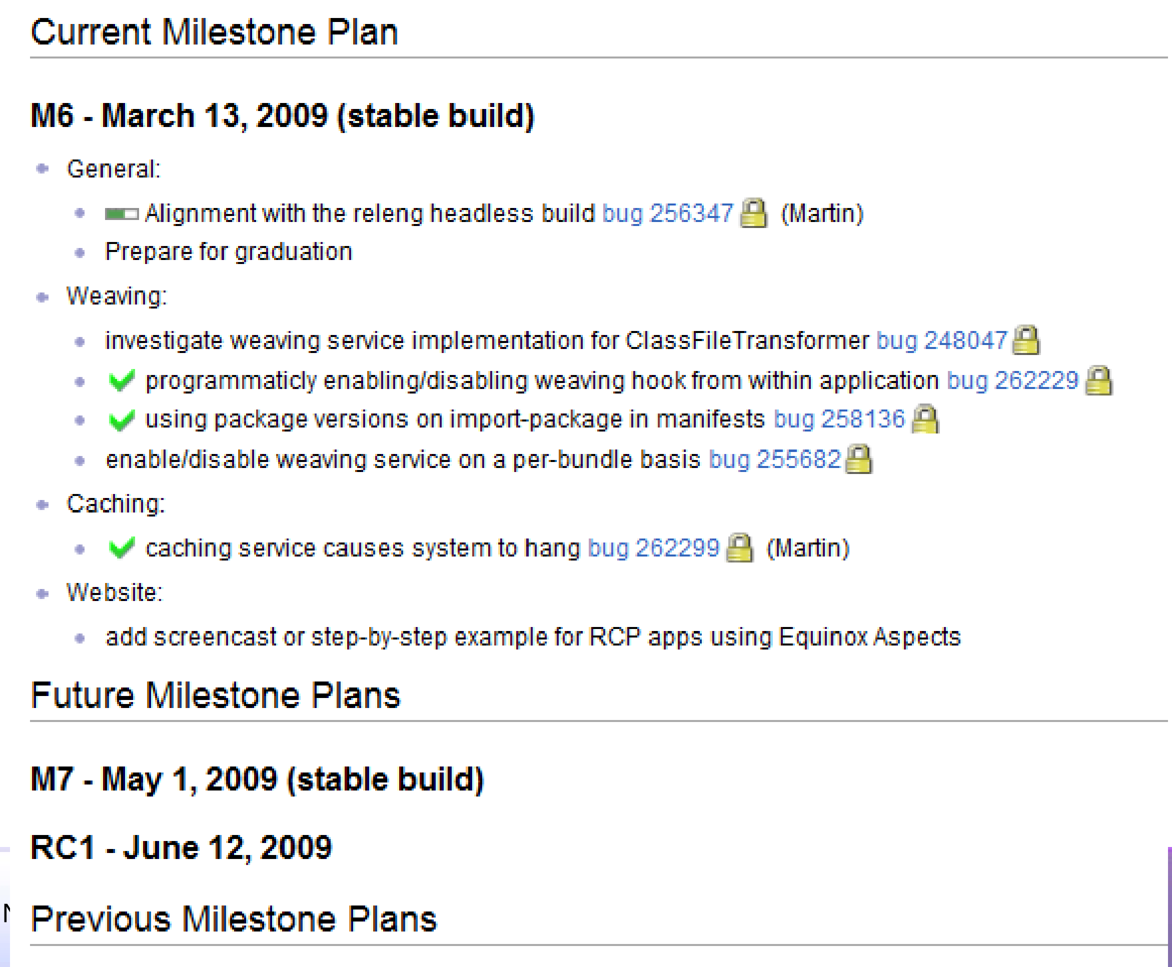
Step 1: Setting up the target platform

```
org.eclipse.equinox.weaving.demo.target
  bundles-1.0.0-3.4.0-1.6.1
    org.aspectj.runtime_1.6.1.20080703120000
    org.aspectj.weaver_1.6.1.20080703120000
    org.eclipse.equinox.weaving.aspectj_1.0.0.200807082136.jar
    org.eclipse.equinox.weaving.caching_1.0.0.200807082136.jar
    org.eclipse.equinox.weaving.caching.j9_1.0.0.200807082136.jar
    org.eclipse.equinox.weaving.hook_1.0.0.200807082136.jar
    org.eclipse.osgi_3.4.0.v20080605-1900.jar
  equinox-aspects.target
```

Transferring data from www.wuetherich.com...

Aligned development + wiki-based planning

- http://wiki.eclipse.org/Equinox_Aspects_Plan



Current Milestone Plan

M6 - March 13, 2009 (stable build)

- General:
  - Alignment with the releng headless build bug 256347 🔒 (Martin)
  - Prepare for graduation
- Weaving:
  - investigate weaving service implementation for ClassFileTransformer bug 248047 🔒
  - ✓ programmaticly enabling/disabling weaving hook from within application bug 262229 🔒
  - ✓ using package versions on import-package in manifests bug 258136 🔒
  - enable/disable weaving service on a per-bundle basis bug 255682 🔒
- Caching:
  - ✓ caching service causes system to hang bug 262299 🔒 (Martin)
- Website:
  - add screencast or step-by-step example for RCP apps using Equinox Aspects

Future Milestone Plans

M7 - May 1, 2009 (stable build)

RC1 - June 12, 2009

Previous Milestone Plans

What‘s Next?

- Ease of use
- Better tooling
- Management API
- Example for another weaver implementation

- Graduation
- First “official” release

Thank you for your attention!

- Questions and feedback welcome!

  http://www.eclipse.org/equinox/incubator/aspects/

- Many thanks to Heiko Seeberger, Andrew Eisenberg and Andy Clement

Martin Lippert: lippert@acm.org