Shake that FUD

How to migrate your Eclipse 3 legacy code to Eclipse 4
Wim Jongman
Nebula, E4 Tooling, ECF
ALM AS/400
Industrial-TSI and Remain Software

Lars Vogel
Eclipse committer
Eclipse and Android developer
Company founder
What is today's story

- Act 1: Why adopt Eclipse 4 RCP?
- Act 2: Create a plan of attack
- Act 3: Execute
Eclipse 3.x deprecated

Eclipse 4 is now officially the new programming model at the Eclipse foundation. While Eclipse 3.x can still be used it is no longer supported. Companies like Remain Software and vogella immediately switched to the new platform and saved significant money in the long run.
Eclipse 4 RCP is better

Flexible
Dependency injection
Extendable
What to migrate

- Core and Util
- User Interface
- Server stuff
What to migrate

Mixed mode

● Allows to use Eclipse 3.x plug-ins in Eclipse 4 applications
Let's not use a tiny example but a real one.
ECF Salvo newsgroup reader

- 32 bundles
- 2 user interface bundles
- Development started 2010
- 5 committers
- 85275 lines of code (says git)
Act 2: Plan of Attack

PRODUCTION

Shake that Fud

DIRECTOR

Vogel/Jongman

DATE

25/3

SCENE

Act. 1: Plan

TAKE

1

CAMERA

That same guy

fodey.com
Options

- Do as little as possible
- Embrace the new programming model
- Use a combined approach
Compatibility mode

You don't change a single bit!

(if you've been a good citizen*)

* Except for the internal API which you illegally used for so many years
This does not feel natural
Steps

- Do as little as possible
- Embrace the new programming model
- Use a combined approach
Things that are replaced

- EditorPart
- ViewPart
- AbstractHandler
- HandlerUtil
- Actions
- Extension based UI contributions
- Advisor classes
- Old Theming
- AbstractUIPlugin
Things that are replaced

- EditorPart > POJO + MInputPart & MDirtyable
- ViewPart > POJO + MPart & MDirtyable
- AbstractHandler > POJO + @Execute
- HandlerUtil > DI
- Actions > POJO commands
- Extension based UI contributions > Model
- Advisor classes > Model
- Old Theming > CSS
- AbstractUIPlugin > Plugin
So in summary....

- POJOs
- Dependency injection
- Application model
Happy!
Happy?
Act 3: Execute
Steps

• Do as little as possible
• Embrace the new programming model
• Use a combined approach
Useful things that are not yet converted

- Wizard Integration
- Navigator FrameWork
- Properties View
- Forms
- DecoratorManager
- IActionFilters
- Preference Pages
- Progress report
- Help system
- The useful bits of org.eclipse.ui
There must be a way out....
Mixed mode
Mixed mode

- Allows to use Eclipse 3.x plug-ins in Eclipse 4 applications
- Unexplored Territory
- Use the model to describe the layout
- Use the model to address 3.x parts
Mixed mode

Shared Elements
- Area
- Part - Properties
- Part - News Sources
- Part - MessageView
- Part - Reply
- Part - New Article
- Part - News Digest View

TrimBars
- Window Trim - Bottom
- Window Trim - Top
Mixed mode
POJOs in Eclipse 3.x with the e4 bridge
A smoother path to migration

- Use the e4 bridge of the e4 tools project to use POJOS in Eclipse 3.x
- Allows a migration view by view, handler by handler and editor by editor
- Once you migrated almost every UI you can switch completely
Summary

- You have an easy way with compatibility layer
- You can migrate full way
- You can use existing 3.x plug-ins
- You can also migrate step by step
What more

- Talks
  - Running the 3x API efficiently on 4 (Paul Webster)
- Blogs we referenced
  - So you are using internal API (Tom Schindl)
  - Using mixed mode (Dirk Fahland)
  - Converting from Eclipse 3 (Lars Vogel)
  - Why Eclipse E4 (Wim Jongman)
- RCP 4 Training
  - Industrial-TSI (Wim Jongman)
  - vogella (Lars Vogel)
Thank you

Sources: https://github.com/ECF/Newsreader

Lars.Vogel@gmail.com

http://www.vogella.com
Twitter http://www.twitter.com/vogella
Google+ http://gplus.to/vogella

Wim.jongman@remainsoftware.com

ALM: http://remainsoftware.com
E4 Training: http://industrial-tsi.com
Twitter http://twitter.com/wimjongman