Crash course on Eclipse

Yann-Gaël Guéhéneuc
guehene@emn.fr

Outline
- Platform
  - Motivation
  - Basic Services
- Java Development Tooling
- Plug-in Development Environment
- What’s next?

Motivation
- Extensible platform
  - Rich, reusable
  - Consistent behaviour, UI
- Guaranteed reliability
  - Open Source
- Promoted innovations
  - Plug-ins
  - No need to reinvent the wheel…
And so what?

- For your developments
  - Specialization training
  - Course-specific projects
  - Master thesis
- For your developments
  - Java Development Tooling
  - C#, Claire, Prolog, Google, UML…
  - AspectJ, CVS, Jad, JProbe, JProfiler…

Platform (1/3)

- A language-independent platform
  - At first, everything is a resource
- A set of frameworks
  - Basic services
  - Graphic framework
- An Open Source project
  - Common Public License

Platform (2/3)

- Published APIs
  - Developer community
- Plug-in mechanism
  - Extensibility
- Repositories
  - CVS
- Java, SWT, JFace, GEF, EMF…
  - Support, performance
Platform

Eclipse SDK
Platform runtime
Eclipse platform
Workspace
V.C.M.
Help
Workbench
JFace
SWT
JDT
PDE
A third-party tool
Another third-party tool
Yet another third-party tool
Yet again another third-party tool

Basic services

Resources
- Workspace, projects
- Directories, files
Modification
- Deltas
Natures
- Builders
Sharing resources

Basic services

Demo?
Java Development Tooling (1/5)

- Java IDE
  - VisualAge for Java (?)
- Perspectives
  - Java, Java Browsing
  - Debug...
- Views
  - Package explorer, Hierarchy
  - Outline...

JDT (2/5)

- Incremental compilation
- Refactorings
- Extension points
  
  [link](http://download.eclipse.org/downloads/documentation/2.0/html/plugins/org.eclipse.jdt.doc.isv/topics_Reference.html)

JDT (3/5)

- Multiple
  - Java SDK
  - Which are the required libraries?
  - Order?
  - Export?
- Which projects does this project depend on?
- In the project, where is the source?
JDT (4/5)

- Multiple JVM
- Applet, Program, JUnit, Eclipse
- Runtime information
- Runtime arguments
- Runtime class-path

JDT (5/5)

Demo?

Plug-in Development Environment (1/5)

- Everything (almost) is a plug-in
  - Lazy loading and initialization
- Controlled extensibility
  - Explicit dependencies
  - Extension point definitions
  - Extension point addition – not replacement
- Extensions of Plug-ins
  - Fragments
PDE (2/5)
- Runtime
- Workspace (resources)
- Workbench
- Team
- Help
- Many more


PDE (3/5)
- Nature
  - Java
  - Plug-in

A plug-in project
A sub-folder containing the source
The plug-in manifest file
The PDE handles the class-path

PDE (4/5)
Plug-in identification, version, main class, ...
Dependencies, dynamic classpath
Local classpath
Extension points extended by this plug-in
What’s next?

- Eclipse as your *incremental* and *integrated* development environment
- Eclipse as a target for your developments

*Let’s go make some plug-ins!*

That’s all folks!

- Thank you so much for your attention!

- Questions?
- Comments?