IDE’s for Java, C, C++

David Rey - DREAM
Overview

• Introduction about IDE’s
  • What is an IDE
  • What is not an IDE
  • IDEs examples for java
  • IDEs examples for C++
  • Eclipse example: overview

• Eclipse demo
  • Project “hello world”
  • Edition facilities
  • Build: compilation and library creation
  • Settings (eclipse and project)
  • Other development tools
    – Doc extractor
    – Debugger
    – The rest: cvs, junit, …
• How to install and use specific plugins: example with a C++ editor plugin
Overview

- Introduction about IDE’s
  - What is an IDE
  - What is not an IDE
  - IDEs examples for java
  - IDEs examples for C++
  - Eclipse example: overview

- Eclipse demo
  - Project “hello world”
  - Edition facilities
  - Build: compilation and library creation
  - Settings (eclipse and project)
  - Other development tools
    - Doc extractor
    - Debugger
    - The rest: cvs, junit, …
  - How to install and use specific plugins: example with a C++ editor plugin
What is an IDE

- IDE = Integrated Development Environment
- IDE = EDI (Environnement de Développement Intégré) in French

- Typical integrated development tools:
  - editor (with auto-indent, auto-completion, colorization, …) ;
  - version control ;
  - compiler/builder ;
  - documentation extractor ;
  - debugger ;
  - testing tools ;
  - refactoring tools.

- Generally language specific
  (c/c++ specific IDE, java specific IDE, not yet good ones for Fortran)

- Example: Eclipse for java (now plugins for C/C++, python, …)
What is not an IDE

- Just a great (complicated?) text editor
- A code generator
- A GUI designer
- A forge (i.e. GForge)
IDEs examples - Java

- Eclipse (http://www.eclipse.org)
- NetBeans (http://www.netbeans.org/)
- JCreator (http://www.jcreator.com/)
- ...


IDE’s examples – C/C++

- Visual C++ - com. license (http://msdn.microsoft.com/visualc)
- Eclipse(CDT)/EasyEclipse (http://www.easyeclipse.org/site/home/)
- Quincy (http://www.codecutter.net/tools/quincy/)
- Anjuta (http://anjuta.sourceforge.net/)
- KDevelop (http://www.kdevelop.org/)
- Code::Block (http://www.codeblocks.org/)
- BVRDE (http://bvrde.sourceforge.net/)
- RHIDE (http://www.rhide.com/)
- ...
Eclipse software

- Foundation: [http://www.eclipse.org](http://www.eclipse.org)
- Common Public License (CPL):
  - royalty free source code / world wide redistribution rights
- Supported languages: Java, C++, python, …
- Supported platforms: Linux, Windows, … (java software)
- Many tutorials on the web
  - A good one (in French):
Eclipse “philosophy”

• Extensible with java plug-ins:
  • May be used with several programming languages via plug-ins (C++, python)
  • Many other add-ons via plug-ins
      (Eclipse3.2 only using mylar plugin to be installed: [http://www.eclipse.org/mylar/](http://www.eclipse.org/mylar/))
    – xml, soap, …
    – …
• Aim: specific IDE depending on usage/user
  (RCP = Rich Client Platform)
Eclipse main features: demo

- Powerful editing mode
  - automatic indentation
  - auto-completion (classes, methods, …)
  - search/replace + refactoring
  - …

- Compiling/building
  - Compilation
  - Libraries creation (e.g. jar files)

- Eclipse/project settings

- Integrated tools
  - Debugger
  - Doc extractor (javadoc)
  - Other tools: cvs, junit
  - …

- Plugins
Overview

• Introduction about IDE’s
  • What is an IDE
  • What is not an IDE
  • IDEs examples for java
  • IDEs examples for C++

• Eclipse demo
  • Project “hello world”
  • Edition facilities
  • Build: compilation and library creation
  • Settings (eclipse and project)
  • Other development tools
    – Doc extractor
    – Debugger
    – The rest: cvs, junit, ...
  • How to install and use specific plugins: example with a C++ editor plugin
Conclusion

IDEs:

- Exist for many platforms and programming languages
  - Java (e.g., Eclipse)
  - C/C++
- Are useful for beginners as well as for experts
- Are time consuming at the beginning
  - Learning curve
  - Several tools need configuring
- Save time when often used:
  - Powerful editing mode
  - CVS/SVN
  - Easy compiling/building/running, related to source files
  - Easy refactoring
- Still in evolution and will be used in the future for
  - RCP
  - GUI designer
  - MDA approach
Screen shots…
Editing with Eclipse: main functionalities
Configuring CVS access

- Check compatibility with CVS version

- Window
  - Open Perspective → Other

- In CVS Repository
  - Click on “add cvs repository”
Configuring compiling/building

- By default, an Eclipse project is compiled internally
- Project → Properties
Using compiling/building: demonstration
Using a build.xml

• It is possible to use ant with xml build files
  • For specific build tasks
  • To set a given version of the compiler, javadoc, …
  • To set specific parameters for building (i.e., on a shared project with people not under Eclipse, …)
  • Cf. Ant and Eclipse tutorial: http://perso.wanadoo.fr/jm.doudoux/java/dejae/chap009.htm#chap_9
Configuring javadoc

• Give path to javadoc at the first go
Configuring javadoc (2)
Generating javadoc: demonstration
Configuring run/debug
Configuring JUnit

- JUnit is a part of the Eclipse project by default
- For more details on how to configure and use JUnit in Eclipse look at [http://perso.wanadoo.fr/jm.doudoux/java/dejae/chap010.htm#chap_10](http://perso.wanadoo.fr/jm.doudoux/java/dejae/chap010.htm#chap_10)
Tools to re-write code: main functionalities