

Integrated Development Environments

Steven J Zeil

February 13, 2013



Outline

- 1 The Components of an IDE
- 2 IDE Examples
- 3 Eclipse



IDEs

Integrated Develop Environments (IDEs) are software packages that attempt to provide comprehensive support for programming

- and possible other software development activities



Outline I

1 The Components of an IDE

2 IDE Examples

3 Eclipse



The Components of an IDE (minimal)

What's the minimum that we expect in an IDE?

- editor



The Components of an IDE (minimal)

What's the minimum that we expect in an IDE?

- editor
- build



The Components of an IDE (minimal)

What's the minimum that we expect in an IDE?

- editor
- build
 - maybe no more than compiler invocation



The Components of an IDE (minimal)

What's the minimum that we expect in an IDE?

- editor
- build
 - maybe no more than compiler invocation
 - with error messages captured/interpreted/walked by editor



The Components of an IDE (minimal)

What's the minimum that we expect in an IDE?

- editor
- build
 - maybe no more than compiler invocation
 - with error messages captured/interpreted/walked by editor
- run/execute



The Components of an IDE (minimal)

What's the minimum that we expect in an IDE?

- editor
- build
 - maybe no more than compiler invocation
 - with error messages captured/interpreted/walked by editor
- run/execute
- debugger



The Components of an IDE (optional)

What would we like to see in an IDE?

- syntax highlighting & aid in editor



The Components of an IDE (optional)

What would we like to see in an IDE?

- syntax highlighting & aid in editor
- documentation (API) look-up



The Components of an IDE (optional)

What would we like to see in an IDE?

- syntax highlighting & aid in editor
- documentation (API) look-up
- flexible/configurable build



The Components of an IDE (optional)

What would we like to see in an IDE?

- syntax highlighting & aid in editor
- documentation (API) look-up
- flexible/configurable build
- packaging/deployment options



The Components of an IDE (deluxe)

What makes us giddy in an IDE?

- smart feedback in the editor



The Components of an IDE (deluxe)

What makes us giddy in an IDE?

- smart feedback in the editor
 - learns API of new code



The Components of an IDE (deluxe)

What makes us giddy in an IDE?

- smart feedback in the editor
 - learns API of new code
 - suggestions



The Components of an IDE (deluxe)

What makes us giddy in an IDE?

- smart feedback in the editor
 - learns API of new code
 - suggestions
- coding aids in editor



The Components of an IDE (deluxe)

What makes us giddy in an IDE?

- smart feedback in the editor
 - learns API of new code
 - suggestions
- coding aids in editor
 - templates



The Components of an IDE (deluxe)

What makes us giddy in an IDE?

- smart feedback in the editor
 - learns API of new code
 - suggestions
- coding aids in editor
 - templates
 - common refactoring (transformations)



The Components of an IDE (deluxe)

What makes us giddy in an IDE?

- smart feedback in the editor
 - learns API of new code
 - suggestions
- coding aids in editor
 - templates
 - common refactoring (transformations)
- documentation generation



The Components of an IDE (deluxe)

What makes us giddy in an IDE?

- smart feedback in the editor
 - learns API of new code
 - suggestions
- coding aids in editor
 - templates
 - common refactoring (transformations)
- documentation generation
- test integration



The Components of an IDE (deluxe)

What makes us giddy in an IDE?

- smart feedback in the editor
 - learns API of new code
 - suggestions
- coding aids in editor
 - templates
 - common refactoring (transformations)
- documentation generation
- test integration
- integration with version ctrl



Outline I

1 The Components of an IDE

2 IDE Examples

3 Eclipse



emacs

The *nix swiss army knife of editors, *emacs* has long functioned as a basic IDE:

- syntax-highlighting editor
- build support (invokes *nix **make**)
 - parses error messages from compilers & other tools
- debugger interface
- works directly with many version control systems



emacs Strengths and Weaknesses

- highly portable
- supports virtually any language you would have a compiler for
- even in windowed mode, leans toward keyboard rather than mouse
 - (not sure if that's a pro or a con)
- outdated interface
- high learning curve



Microsoft Visual

Visual Studio

- syntax-highlighting editor
 - background compilation provides quick feedback on simple errors
- built-in build manager
 - limited configurability
- debugger interface
- some designer tools (e.g., design classes in UML)



Visual Strengths and Weaknesses

- wide variety of languages (but Microsoft processors)
- single-OS
- closely integrated with Microsoft compilers
- modern, mouse-oriented interface



Visual Strengths and Weaknesses

- wide variety of languages (but Microsoft processors)
- single-OS
- closely integrated with Microsoft compilers
- modern, mouse-oriented interface
 - What will Windows 8 do to that?



NetBeans

Free IDE originally distributed by Sun as “the” development platform for Java.

- Still largely Java centric, though some support for other languages
 - particularly web-related languages like Javascript, CSS, XSL
- Portable (written in Java)
- Tends to track the trends and hot topics in the Java world promptly
- editor, build manager, debugger
- moderately extensible



Single-Language IDEs

The open source community has produced numerous single-language IDEs.

Many are focused on educational use.

Examples:

C++ Bloodshed Dev-C++, Code::Blocks

Java BlueJ, Dr. Java, jGrasp



Outline I

- 1 The Components of an IDE
- 2 IDE Examples
- 3 Eclipse**



Eclipse

Probably the hottest IDE in the open source world:

- syntax-highlighting editor, multi-language support
 - strong hinting, API, interface aid
 - templates and refactoring



Eclipse

Probably the hottest IDE in the open source world:

- syntax-highlighting editor, multi-language support
 - strong hinting, API, interface aid
 - templates and refactoring
- build support
 - easily configured or switched to other build tools



Eclipse

Probably the hottest IDE in the open source world:

- syntax-highlighting editor, multi-language support
 - strong hinting, API, interface aid
 - templates and refactoring
- build support
 - easily configured or switched to other build tools
- background compilation for quick detection of language errors



Eclipse

Probably the hottest IDE in the open source world:

- syntax-highlighting editor, multi-language support
 - strong hinting, API, interface aid
 - templates and refactoring
- build support
 - easily configured or switched to other build tools
- background compilation for quick detection of language errors
- integrated *unit testing support
- solid debugger, intuitive handling of threads



Eclipse

Probably the hottest IDE in the open source world:

- syntax-highlighting editor, multi-language support
 - strong hinting, API, interface aid
 - templates and refactoring
- build support
 - easily configured or switched to other build tools
- background compilation for quick detection of language errors
- integrated *unit testing support
- solid debugger, intuitive handling of threads
- some packaging & deployment support



Eclipse

Probably the hottest IDE in the open source world:

- syntax-highlighting editor, multi-language support
 - strong hinting, API, interface aid
 - templates and refactoring
- build support
 - easily configured or switched to other build tools
- background compilation for quick detection of language errors
- integrated *unit testing support
- solid debugger, intuitive handling of threads
- some packaging & deployment support
- integrates with most version control systems
- modular plug-in extensibility with a rich variety available

