Integrated Development Environments

Steven J Zeil

February 13, 2013

Contents

1 The Components of an IDE 3

1
Integrated Development Environments

2  IDE Examples  6

3  Eclipse  13

CS795  2
Integrated Development Environments

IDEs

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

- and possible other software development activities

1. The Components of an IDE

The Components of an IDE (minimal)

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

- editor
Integrated Development Environments

• 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
Integrated Development Environments

- 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
  - learns API of new code
  - suggestions
Integrated Development Environments

- coding aids in editor
  - templates
  - common refactoring (transformations)
- documentation generation
- test integration
- integration with version ctrl

2 IDE Examples

emacs
Integrated Development Environments

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

References, if you are unfamiliar with this:

- Compiling in *emacs*

- *emacs* Debugging mode (CS252)
**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- outdated interface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
Integrated Development Environments

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)
Integrated Development Environments

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?

I’ve never been fond of Visual, but that comes more from my opinion of the MS compilers. MS C++ had recurring issues with basic standards conformance and std library implementation. And MS’s support of Java was perpetually luke-warm.
**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
Integrated Development Environments

Netbeans and Visual clearly stole interface ideas from one another. (Then Eclipse came along and stole from them both.)
I have not used NetBeans in a long time. I remember it as being incredibly sluggish even on reasonably high-powered desktops.
My enduring impression is that Eclipse seemed to do everything NetBeans wanted to do, did it about 6 months later, but did it better.

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

CS795
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

- build support
  - easily configured or switched to other build tools
Integrated Development Environments

- 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

Eclipse is available here.