ErlIDE

A modern GUI for erlang development

Eclipse

- An extensible platform for developing applications
- 100% Open Source
- 100% java
- created by IBM
- Many platforms
  (Windows, Mac, Linux, Solaris)

Why Eclipse?

- Mature framework
- Large ecosystem of useful tools
- A picture is worth a thousand words

- Somewhat bloated and memory hungry. Better with Java 6.
- Impedance mismatch with Erlang

ErlIDE

- A set of plugins for Eclipse
- Project started in 2000, restarted in 2002 and getting ”real” from 2005.
- Mainly 2 developers, we’d like to see more!

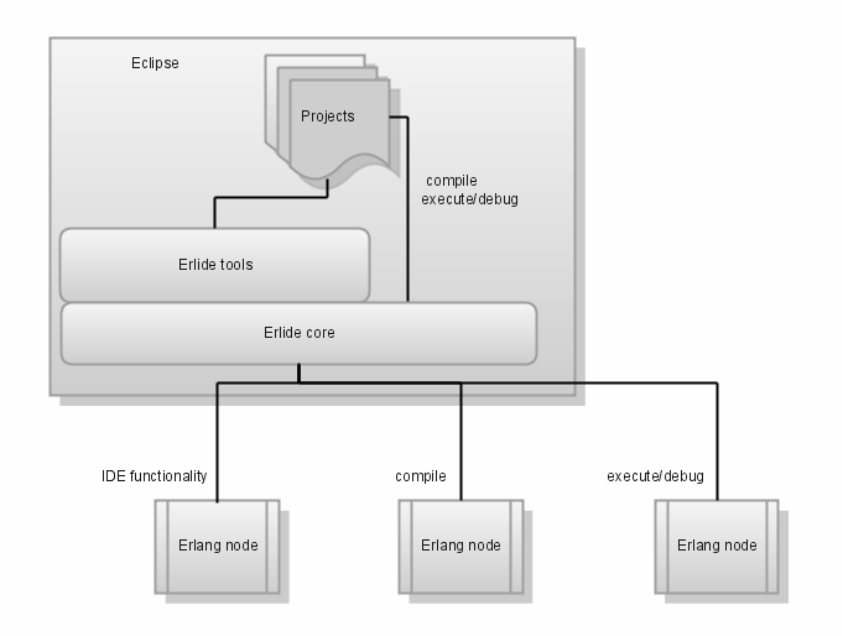
ErlIDE

- Originally 100% Java
- Code: 80% Java, 20% Erlang
- Functionality: 50% java, 50% Erlang

Architecture

- Eclipse workspace uses an Erlang backend to do the work
- Each project may be compiled by a separate backend
- Another backend is used to run the projects

Architecture



Architecture

- Erlang side provides services to Java side
  - RPC
  - event based
- Infrastructure to make communication as seamless as possible
  - implement Java interfaces in Erlang

ErlIDE and OTP

- Requires R11B-5 or later
- Special versions of jinterface, scanner, parser, syntax_tools, edoc, debugger

Features

- Editor

•Indentation of Erlang code

•Syntactic highlighting

•Bracket matching

•Automatic completion of erlang functions and records

•Selective display of functions and declarations (folding)

- Indentation and code formatting

•Automatic indentation

•Formatting and pretty-printing

Features

- Hover
  - OTP documentation shown for external calls
  - Show declaration of macros and records
- Outline
  - Code outline of the structure of an Erlang module
  - Filtering of Erlang functions and declarations
  - Quick outline for selecting Erlang function or declaration

Features

- Navigation
  - Go to declaration of function, macro or record
- Project Import
  - Erlang-aware import of projects
- Creation wizards
  - Create Erlang projects
  - Create Erlang modules with code skeletons

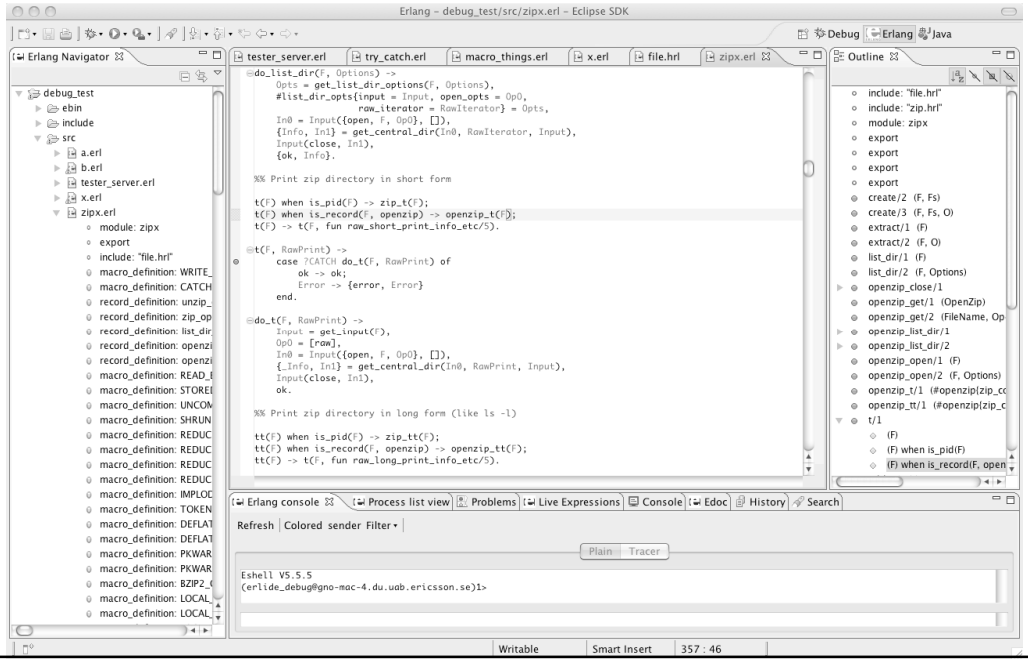
Features

- Debugger
  - Erlang debugger within eclipse debugger framework
  - Breakpoints, single-stepping
  - Inspection and modification of local variables
  - Distributed debugger, debug on multiple nodes

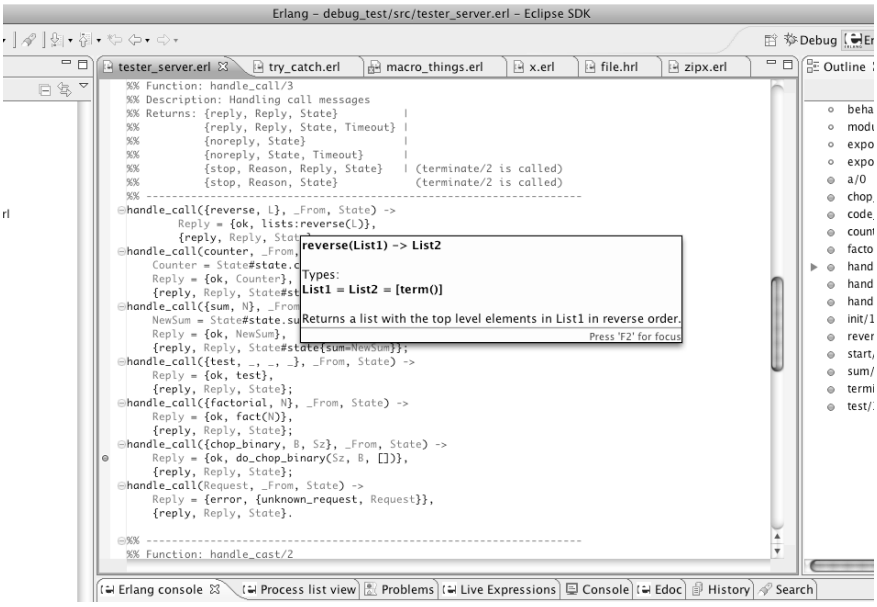
Features

- Erlang runtime handling
  - Provision for different runtimes, local or remote
  - Develop on one runtime, test and debug on another
- Warnings, errors, TO-DO-markers
  - Show errors and warnings in code and in problems view
  - Automatically mark and list comments with TODO and FIXME
  - Erlang log printouts with link to code line

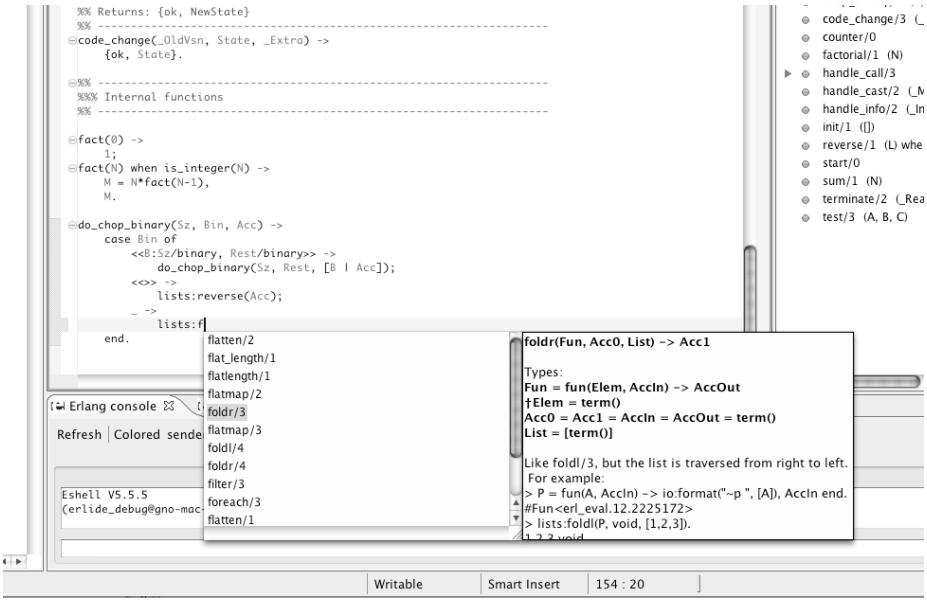
Demonstration



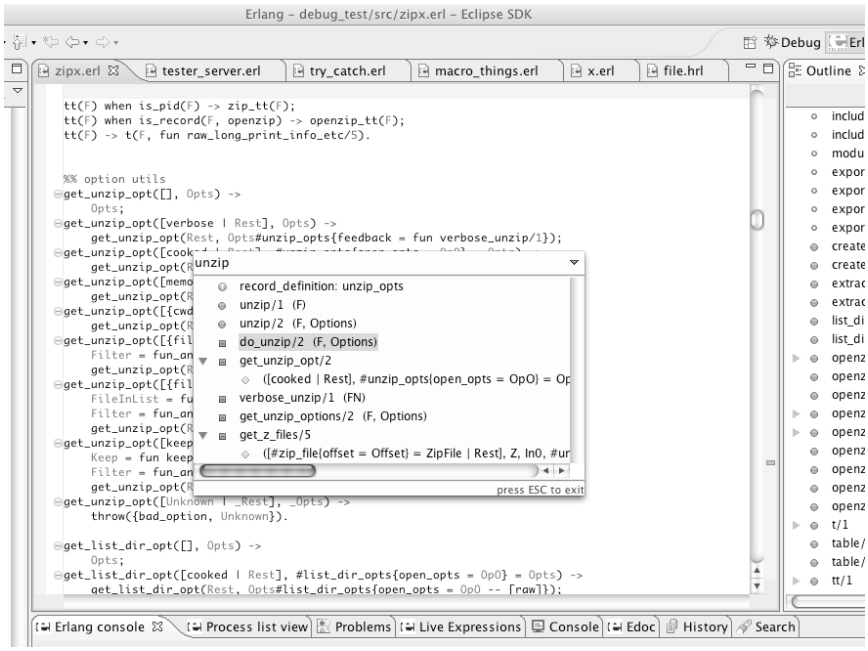
Demonstration



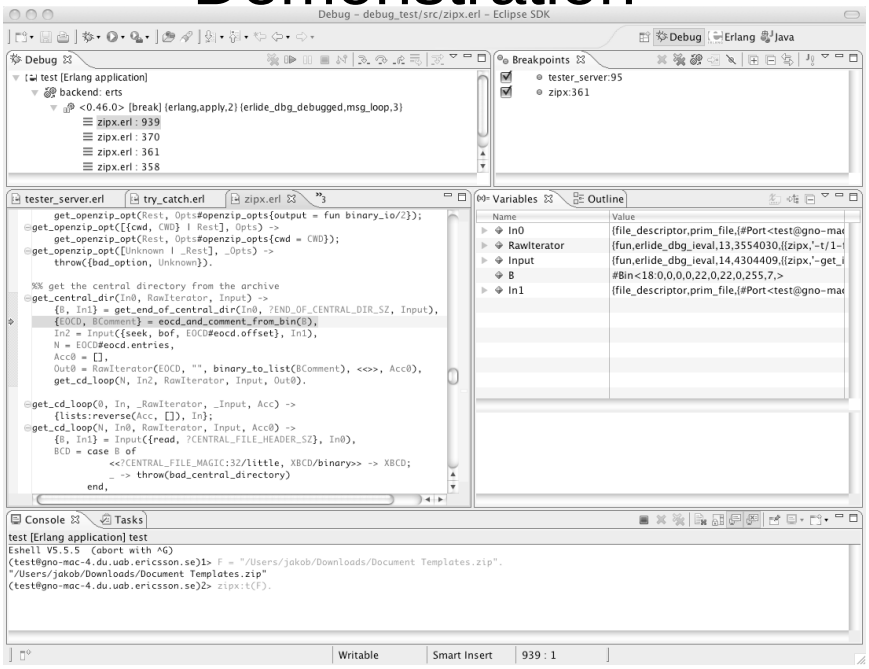
Demonstration



Demonstration



Demonstration



Some future features

- Semantic highlighting in editor (function calls, variable usage, etc)
- Show documentation of user functions
- Erlang-aware search
- Refactoring
- xref-based indexing and searching
- Tracing viewer
- Syntax error correction
- What tools do <u>you</u> need?

Resources

- http://erlide.sourceforge.net
- http://erlide.sourceforge.net/update

- http://eclipse.org
- http://erlang.org