

# The Scala IDE for Eclipse

## Retrospect and Prospect for 2.8 and Beyond

Miles Sabin, Chuusai Ltd.  
<http://www.chuusai.com/>

<http://uk.linkedin.com/in/milessabin>  
<http://twitter.com/milessabin>

# Outline

- History
- Current Work
- Demo
- A Call to Action!

# History

# History 1

- Started early in Scala's timeline (early 2005)
- Very simple IDE plugin
- Little functionality beyond basic syntax highlighting and build invocation
- Written in Java
- There was a least one other similar offering at the time: Scaliptor

# History 2

- Announced December 2005, first release February 2006 (2.1.0)
- Rewritten in Scala
- Some semantic features acquired (eg. limited auto-completion)
- Even at this early stage requests for the ability to mix Java and Scala and JDT interop were coming in

# History 3

- Announced June 2007, first release February 2008
- Attempted much deeper integration with the Scala compiler,
  - Interactive error reporting
  - Semantic highlighting
  - Incremental compilation
  - Dependency management
- Many hooks added to scalac

# History 4

- Start of my involvement (May 2008)
- Prompted by very generous sponsorship by EDF Trading
- Primary goals
  - Ease Java/Scala migration
    - Mixed Scala/Java compilation in scalac
    - Mixed Scala/Java projects in Eclipse
  - Improved Eclipse stability and release process

# History 4

- First commit to trunk in July 2008
- Results of the work first visible in 2.7.2.RC1 in August 2008
- Final 2.7.2 release in November
- Goals somewhat met,
  - Mixed Scala/Java enabled in scalac and Eclipse
  - Release process dramatically improved
  - However JDT integration limited and stability issues remain



# History 5

- Serious roadblocks to JDT integration
- 2.7.3 contained only very minor bugfixes and enhancements
- Long standing patch to open up the JDT for extension updated but rejected
- AspectJ and Equinox Aspects provided a Plan B

# History 5

- JDT Roadblocks removed enabling rapid progress,
  - Many bugs fixed
  - Many features enhanced
- First appeared in 2.7.4 in April 2009
- Some drawbacks however,
  - Some installation issues
  - Some JDT features magically started to almost work

# History 6

- Solution to preceeding: full speed ahead on JDT integration, recruit contributors
- However, much of the codebase complex over-abstracted and forbidding
  - Learning curve too steep for casual contribs
  - Fragility in presentation compiler
    - Spurious pink squiggles, minor errors destroy all highlighting, unreliable dependency management
  - Negative effect on scalac
    - Many fragile “if (inIDE) ...” blocks

# History 6

- Complete reimplementation of all semantic and build related features
- Work done in collaboration with Martin Odersky and Iulian Dragos at EPFL in May last year

# History 6

- New interactive compiler
  - Completely replaces old presentation compiler
  - Now the unit of (re)compilation is a whole source file.
    - Performance is on a par if not better than before, old approach a “premature optimization”
  - All “inIDE” conditions removed from scalac
  - Accurate position information to AST nodes
    - Enables new features (folding)
    - Supports new tools (refactoring, formatting)

# History 6

- New incremental build manager
- Far more reliable than previous Eclipse-specific code
- Makes use of scalac's filesystem abstraction enabling,
  - Use by other IDEs
  - Use by other tools
- Supports all JDT build path features
  - Multiple source and output directories

# History 6

- Hubert Plociniczak at EPFL now maintaining the build manager
- Many useful contributions from Mark Harrah

# History 6

- Drawbacks
  - Big bang ... there's no going back,
    - All support for previous integration has been removed from scalac
    - All semantic features in Eclipse had to be reworked
    - No feasible backport of new features or bugfixes to 2.7.x branch
  - This is in stark contrast to the incremental improvement policy I had intended to adopt from 2.7.2 onwards.



# Current Work

# Current Work

- Hyperlink navigation (“Jump to definition”) close to completion
  - Completely new implementation based on the JDT's mechanisms
  - Scala source code selections mapped to Scala AST nodes and Eclipse JDT model handles
  - Some “re-sugaring” required to map AST nodes to appropriate JDT construct
  - Supports “Jump to implementation” in many cases

# Current Work

- The underlying mechanism supports additional JDT features,
  - Javadoc hovers
  - Source hovers
  - Java model search (eg. find references, find annotations)

# Current Work

- Reusing the JDTs navigation required the extension of AST mapping to binaries (eg. class files in the Scala standard library Jars)
  - Currently only supported for binaries with source attachments
- This provides dramatic improvement in navigability for the standard library
  - Library components now open in Scala editor
  - Outline view and outlines in package explorer now available

# Current Work

- The full mapping of Scala source and binary components enables comprehensive JDT-wide indexing of Scala elements
  - The JDT indexer now understands Scala traits and objects
  - Indexing supports uniform “Open Type” behaviour across Scala and Java projects
  - Supports “Find references” to both Scala & Java in both Scala & Java sources
    - Still work in progress, but already useful

# Current Work

- The new presentation compiler is much more reliable
  - No longer used for syntax highlighting, so doesn't have to run synchronous with keystrokes and whole CU update is fast enough
- Currently supports,
  - Error reporting as-you-type
  - Live outline update in the outline view and the package explorer
  - Code completion

# Current Work

- New and very valuable collaborations
  - Formation of a "JDT hackers" group
    - understandably cautious support from the JDT team
  - Work with Sonatype on Scala support in m2eclipse
  - Integration with Eclipse PDE build tools in 3.6
  - Mirko Stocker's Scala refactoring project
  - Matt Russell's Scalariform formatter

Demo



Who uses an IDE?

Who uses an IDE for Scala?

Who Uses Eclipse?

Who uses Eclipse for Scala?

A Call to Action!

# A Call to Action!

- It's essential to lower the barrier to contribution
- The codebase has been made much less forbidding
- But infrastructure, build, documentation, reporting, communication need more work

A Call to Action!

<http://www.scala-ide.org/>

A Call to Action!

Git!

(at <http://scala-ide.assembla.com/>)



A Call to Action!

Tycho/Maven 3.0!!

# Participate!

- The Scala IDE for Eclipse's home  
<http://www.scala-ide.org/>
- Wiki, git repository, bugs  
<http://scala-ide.assembla.com/>
- Mailing lists  
<http://groups.google.com/group/scala-ide-user>  
<http://groups.google.com/group/scala-ide-dev>
- Follow @ScalaIDE on Twitter

# The Scala IDE for Eclipse

## Retrospect and Prospect for 2.8 and Beyond

Miles Sabin, Chuusai Ltd.  
<http://www.chuusai.com/>

<http://uk.linkedin.com/in/milessabin>  
<http://twitter.com/milessabin>