Release Review: AJDT 1.6.0 and AspectJ 1.6.0

Andrew Clement
March 19, 2008

Introduction

- Co-ordinated release of:
  - AspectJ 1.6.0
  - AJDT 1.6.0 (embedding AspectJ1.6.0) for Eclipse 3.4
    - plus service refresh of AJDT on Eclipse 3.3 to include AspectJ1.6.0)
- AspectJ also made available separately for command-line and build system use (Ant)

2.1 Features

- AspectJ 1.5 was based on the Eclipse 3.1 compiler

  => it was a 1.5 compiler

- Main goal of AspectJ 1.6 is rebasing it on the Eclipse 3.3 compiler
  - Version 0.785_R33x of jdt.core

  => making AspectJ a 1.6 level compiler

- The version number goes to 1.6 to bring it in line with javac

2.1 Features

- AJDT 1.6.0 is the major update to AJDT that will embed the new AspectJ inside the Eclipse environment
- AspectJ bug fixes that have gone during 1.6.0 development
  - Bugzilla Query: http://tinyurl.com/256xdl
  - Notable areas that received a number of fixes:
    - generics handling
    - load time weaving
    - annotation handling
- Incremental compilation enhancements for a multi-project configuration
  - More frequently incrementally builds the code

  => big productivity help, see numbers shown in http://dev.eclipse.org/mhonarc/lists/aspectj-users/msg09002.html

2.1 Features – AspectJ language changes

- Annotation value matching
  - Able to statically match on not only whether an annotation exists but on where particular members of the annotation have particular values

```
execution(@Foo(id=”Andy”) * *(..))
```

    - *'Execution of any method annotated with @Foo where the id within the annotation instance is set to 'Andy''*
  - Previously this was only possible using hand written reflection on the annotations

2.1 Features – AspectJ language changes

- Parameter annotation matching
  - Previously ignored by AspectJ and could not be matched upon
  - AspectJ1.6 supports matching based on parameter annotations, in addition to parameter type annotations. Example method and matching pointcut:

```
public void foo(@ParamOne String p1) {}
execution(* *(@ParamOne (*)))
```

  - *'Execution of any method with a single parameter annotated with @ParamOne'*

2.2 Non-Code aspects

- No major documentation changes in this release
  - Minor updates to cover new syntax
- Examples and tutorials are still correct and up-to-date

2.3 APIs

- APIs as exposed from the underlying JDT compiler have changed in line with the move to the new 3.3 JDT compiler
  - For example, AST API
- No changes to APIs exposed directly from AspectJ/AJDT
  - Compilation invocation, feedback on weaving, AspectJ code parsing

2.4 Architectural Issues

- AspectJ/AJDT
  - Large number of code changes to rebase on 3.3 compiler, but:
    - Exact same approach taken as in previous releases
- AspectJ is a well thought out series of extensions made to the JDT core compiler
  - Not specific to Aspect Oriented Programming (AOP) concepts
  - Extensions must be well designed to avoid fragility when moving to a new compiler level (this proved to be true in the move to 3.3!)
- The AspectJ weaver is still rather heavy on memory usage
  - Was not really a problem until load-time weaving became popular, since one weaver is created per classloader and some application server environments have 100s of classloader instances

    ⇒ Will get much more focus once stable 1.6 compiler released

2.5 Tool Usability

- The tools are mature and have been used in a number of production environments already

2.6 End-of-Life

- No APIs or significant user features from previous releases are being end-of-life'd in this release

2.7 Bugzilla

- Focus for AspectJ 1.6.0 is becoming a 1.6.0 compiler
  - Priority is to bugs related to being 1.6.0 compliant and regressions since the previous release (1.5.4)
  - Target for 1.6.0 – no known regressions
- AspectJ since previous 1.5.4 release (December 2007):
  - 40 bugs raised (including 2 enhancements)
  - 62 bugs resolved (including 5 enhancements)
  - For 1.6.0RC1 – target will be no known regressions, no priority 1s or 2s open against 1.6.0
- AJDT since previous 1.5.1 release (January 2008):
  - 8 bugs raised
  - 1 bug resolved (Java 6.0 compliance)

  => Effort has been focused on AspectJ 1.6.0

2.7 Standards

- J2SE
  - AJDT runs on J2SE 1.4 and 1.5 and 1.6
  - The AspectJ compiler will run on JS2E 1.4 and higher
    - Move to 1.4 from 1.3 is due to Eclipse compiler move to 1.4
  - The code created by the AspectJ compiler will run on J2SE 1.1 and higher

2.8 UI Usability

- Accessibility:
  - No review for this release, but a previous major version was given an accessibility review and all issues found then have been resolved since (http://www.eclipse.org/ajdt/accessibility1_3.html)
- We follow the User Interface Guidelines

2.9 Schedule

- AspectJ1.6.0 has stuck to its planned schedule
  - Milestone 1 mid-January
  - Milestone 2 mid-February
  - RC/Final towards the end of March

  => http://eclipse.org/aspectj/plans_new.php

- AJDT is the main route for users to consume AspectJ
  - AJDT release intended same day as AspectJs release
    - On both Eclipse 3.3 and 3.4. Eclipse 3.2 builds are now not done unless critical fixes need backporting

Process

- These releases been developed using open, transparent, permeable, and inclusive processes
- Use of Bugzilla, the AJDT newsgroup, the AspectJ users mailing list, and the developer mailing lists
- Builds are done using the Eclipse provided infrastructure with cruisecontrol for continuous integration

2.10 Communities

- Continuing to foster active community:
  - Regular monitoring of AJDT newsgroup, AspectJ users mailing list, and Bugzilla
  - Heavily used by Spring – we regularly monitor the Spring AOP related forums too

2.11 IP Issues

- IP process followed
- IP logs:
  - AJDT: http://www.eclipse.org/ajdt/project-info/ip-log.txt
  - AspectJ: http://www.eclipse.org/aspectj/project-info/ip-log.txt
- Released under EPL
- No new external components included in these releases
  - Apart from Eclipse 3.3 compiler now embedded from jdt.core

2.13 Project Plan

- Future releases:
  - AspectJ 1.6.1
    - Memory usage and load time weaving enhancements planned
  - AJDT 1.5/1.6 refreshes for Eclipse 3.3 and 3.4 final
  - AspectJ plan is as per documented at:
    - http://www.eclipse.org/aspectj/plans_new.php

Release Review Version

- These slides are based on the following version of the Release Review document:
  - http://www.eclipse.org/projects/dev_process/release-review.php December 10, 2007