Dali Project – 0.5 Release Review

21 June 2006

Neil Hauge

Review Topics

- What is Dali?
- Who is involved?
- When are the relevant milestones?
- Feature Overview
- Standards Supported
- IP Review
- API Overview
- Documentation Status
- Testing Status
- Q&A

What is Dali?

- Overview
  - The goal of the Dali JPA Tools Project is to build extensible frameworks and exemplary tools for the definition and editing of Object-Relational (O/R) mappings for EJB 3.0 Java Persistence API (JPA) Entities. JPA mapping support will focus on minimizing the complexity of mapping by providing entity generation, validation, and rich UI editing with JPA default information.
  - Dali is currently an incubator project in the WTP
  - This is a pre-1.0 technology release review timed to coincide with WTP's 1.5 release.
  - The Dali Plug-ins will be made available for download on top of the 3.2 / 1.5 platforms via WTP's download site and update manager (separate from 1.5)

Who is involved?

- Project Participants
  - Oracle – 6 Committers, 2 Major Contributors
  - Versant – 3 Committers
  - JBoss – 1 Committer
- Community Interaction
  - Many bug/enhancement reports and code contributions received from community including bug fix and enhancement patches
  - Active newsgroup and mailing list used to promote an open atmosphere
  - Project Status/Milestone Planning calls held once a month – meeting minutes posted on mailing list and also recently documented on the project wiki – http://wiki.eclipse.org/index.php/Dali_Project

Who is involved? (Cont.)

- Community Interaction (Cont.)
  - Project Milestones tracked on the Wiki
  - Bugzilla used for feature tracking and bug tracking
  - Attending weekly WTP-PMC meetings
  - Dali has been presented at JavaPolis 05, OOP 05, EclipseWorld 05, EclipseCon 06, and JavaOne06
  - Reviews
    - Creation Review on July 26th 05
    - Move Review (into WTP) on May 24th 06
    - Passed WTP internal release review on June 7th 06
  - Nearly 6000 Dali downloads since February 06.

When are the relevant milestones?

- 0.5 Release Schedule
  - M3 - May 12th
  - M4/RC0 - June 4th(Feature freeze)
  - RC1 - June 21st (Code freeze)
  - RC2 - June 28th (0.5 Release - dependent on WTP RC6)
  - R0.5 – June 30th, 06 (In coordination with Callisto)

When are the relevant milestones? (Cont.)

- Post 0.5
  - M6: End of August
    - WTP Facet adoption
  - M7: End of September
    - JPA runtime extension support for Java SE projects
  - M9: End of November
    - Greater architectural alignment with Platform including Tabbed Property Panes
  - M10/R1.0?: End of December
    - JPA XML Descriptor (orm.xml) support
    - APT based code completion in the Java Editor (Depends on level of support provided in APT)
    - Provisional API's

Feature Overview

- Persistence Properties View
  - Provides rich UI editing of JPA related metadata
  - Unified editing for JPA annotations and eventually the JPA XML Descriptor (or orm.xml) in M10
  - Provides dynamic default values for JPA metadata
- Persistence Outline
  - Provides visual representation of the elements that make up a Java Persistence Entity
  - Allows navigation of Persistence Properties view (in addition to code navigation)
- Validation
  - Validation in the form of “Problems” are reported based on the state of the Entities in the dali.orm model
  - Validation is also performed on the persistence.xml in the dali.packaging model

Feature Overview (Cont.)

- Entity Generation from Tables
  - Generates spec compliant Entities from tables contained in the WST.RDB database definition model
- Table Generation from Entities
  - Generates platform specific tables from Entities through the DDL generation capabilities of the WST.RDB component
- Tool Usability
  - The JPA tools developed in this project are already being used to assist in the development of applications.

Standards Supported

- EJB3.0 Java Persistence API
  - Dali specifically supports the Java Persistence API (JPA), which is part of the larger EJB 3.0 specification (JSR-220). The EJB 3.0 specification is part of the larger Java EE 5.0 specification (JSR-244).
  - The JSR-220 specification was finalized on May 11th 2006.

IP Review

- About files and license files are complete and correct?: *Yes*
- All significant and third-party contributions have been reviewed by Eclipse Legal? *Yes: See project IP log for complete list*
- Project IP log complete and has been reviewed by Eclipse Legal? *Yes*
- The URL is http://www.eclipse.org/webtools/development/ip_log.html

API Overview

- No Public or Provisional API's have been defined for the 0.5 release. The current plan is to develop public and provisional API's for the 1.0 release based on further input from actual adopter extension requirements.
- Several extensions points have been defined for areas that are clearly going require them, such as adding new types of mappings. These extension points, in addition to likely future extension points, are defined in the developer documentation located here - http://wiki.eclipse.org/index.php/Dali_Developer_Documentation
- Usage of EMF is expected to provide some level of built-in model extensibility
- There are no end-of-life issues as there is no API yet

Documentation Status

- User Documentation
  - Near comprehensive user documentation currently available in Eclipse Help or PDF format. Contributed by professional tech writer at Oracle.
  - Quick Start and near comprehensive Tutorial available
  - Context sensitive help is also provided for nearly all Dali controls
  - Available here - http://www.eclipse.org/dali/docs/dali_user_guide.pdf and in Eclipse Help.
- Developer Documentation
  - Initial documentation available – covers the architecture and describes the plug-ins that make up the Dali JPA Feature.
  - Covers preliminary extension points, and gives brief examples
  - Will act as a guide for further documentation.
  - Available here - http://wiki.eclipse.org/index.php/Dali_Developer_Documentation

Testing Status

- Dali has two sets of JUnit tests, one for the dali.utility plug-in and one for dali.core plug-in
  - The dali.utility plug-in is thoroughly tested to insure proper operation of these core classes (advanced iterators; file, string, collection utilities). There are currently 485 unit tests for this plug-in.
  - The dali.core plug-in tests focus on specific feature functions as well as system level tests, including model initialization, project construction/update/deletion.There are currently 69 tests for this plug-in.
- All tests are currently passing
- The Dali project also has a dedicated QA Engineer who has started writing automated UI tests using TPTP's automated test framework.
- Bug Counts:
  - 59 Open (20 bugs(2 P2's), 39 Enhancements);
  - 88 Fixed (61 bugs (15 P2's), 27 Enhancements)

Questions

- Please visit Dali on the web at www.eclipse.org/dali