Orbit Creation Review

Jeff McAffer
IBM Ottawa Software Lab

Introduction

- Increased use of third party libraries in Eclipse projects
- IP process is being improved to allow approval for use in all projects
- No mechanism for coordination and sharing of bundled versions of approved third party libraries
- This leads to;
  - Duplication of effort
  - Duplication of content in our deliverables/downloads
  - Duplication of classes loaded in memory at runtime
- Counter-productive in a componentized world

Issues

- Inconsistent “bundling” of libraries leads to confusion and duplication
- Third party function (e.g., Tomcat) is often a collection of libraries from fourth parties
- Gathering, componentizing and bundling these libraries is costly
- People using Eclipse often need these libraries as well

Proposal

- Provide a focal point for Eclipse projects to find and/or contribute bundlings of approved third party libraries
- No “development” done in Orbit
- Orbit acts as a hub or distribution point
- Also a place to discuss and track bundling techniques and issues
- All Eclipse projects should, by default, use the bundles in Orbit
  - Release reviews should identify, explain and justify exceptions

Scope

- Orbit does NOT replace or modify the Eclipse IP process
  - Projects must still submit contribution questionnaires as before
- Only the specific versions of specific libraries approved by the Foundation will be managed and distributed by Orbit

Organization

- Components
  - Orbit will have only one component, “Bundles”
  - It is conceivable that we create one component per library (e.g., log4j) included in Orbit but that would result in many components
  - This approach can be revisited if scaling problems are encountered
- Project Team
  - Committers in Orbit will be drawn from committers in other projects
    - In general each Eclipse project that has unique third party library needs will have a small number (e.g., one) Orbit committer
  - New committers can be added as new contributions are “donated”
  - Committers with key knowledge wrt bundling, classloading, etc have signed up for Orbit
  - The project as a whole will be led by Jeff McAffer, IBM.

Community

- Most of the heavy users of third party libraries have pledged support for and interest in contributing to Orbit
    - Eclipse, DPT, TPTP, STP, WTP, GMF, OHF, Dash, DSDP, …
- A few quotes from the newsgroup…
    - “Well done for taking on this vital task … you have our strong support.” – STP
    - “The Orbit project will be very useful in standardizing the naming and versioning of these common plug-ins…” – TPTP
    - “The GMF project is interested in participating…” – GMF
    - “This is a great project idea. I really approve.” – OHF

Conclusion

- Natural evolution of the componentization in Eclipse
- Offers a focal point for consuming third party libraries
- Furthers the Java component model approach that is fundamental to Eclipse's success
- The work is already being done, Orbit makes it easier to share
- The name is cool