



Buckminster

Ganymede Review

May, 2008



© 2008 Cloudsmith Inc - Made available under the Eclipse Public License.

AGENDA

- Buckminster Objective
- Buckminster Capabilities
- Common Components used
- API Quality
- IP Clearance and License
- Committer Diversity
- Suggestions for Next Steps

BUCKMINSTER OBJECTIVE

Buckminster's objective is to leverage and extend the Eclipse platform to make mixed-component development as efficient as plug-in development. The basic approach is to introduce a project-agnostic way of describing a development project's component structure and dependencies, and to provide a mechanism for materializing source and binary artifacts for a project of any degree of complexity.

BUCKMINSTER CAPABILITIES

- **Complex dependency resolution**, providing recursive resolution of dependencies leveraging existing Eclipse "Team Providers," with the addition of new retrievers, for exemplary purposes, covering source and binary artifacts that are not version-controlled in a traditional sense. Resolution uses a variety of versioning schemes and is based on match rules similar to those found in the Eclipse plug-in framework. This allows comparison of current and prior dependency resolutions to support update impact analyses.
- **Uniform component dependency format**, using a component-type agnostic mechanism for describing components and their respective targets and dependency requirements. Most Eclipse projects, and many other component types, have some level of dependency information that can be leveraged. Extensions can be added to provide additional strategies for dependency pattern recognition.
- **Intelligent retrieval mechanisms**, using a component query mechanism the resolution and generated bill of material needed for a given configuration are separated from the actual materialization. This allows sharing of configurations with varying degree of variability between team members (e.g. from "all source needed for latest revision on main branch" to "frozen release configuration").
- **Flexible project workspace binding**, allowing components materialized on disc to be bound to a workspace in different ways, including invoking "build action" before binding to a workspace and supporting "Proxy Projects" consisting of links to physical artifacts and auto-generated Eclipse project information. These capabilities are helpful when sharing code or other artifacts that are not eclipse projects.

BUCKMINSTER CAPABILITIES

- **Actions**, leveraging existing “build technologies” both within Eclipse (PDE-build) and external (ANT), Buckminster can drive building, and assembling components. Materialization is not just “copying of files” – a compiled artifact can be materialized from its source.
- **Headless mode**, a headless packaging of Buckminster gives the same capabilities to command line level tools and scripts as what is available in the Eclipse IDE. As an example Buckminster can drive headless PDE builds.

COMMON COMPONENTS USED

- Eclipse Resources
- Eclipse CVS support
- Eclipse PDE build infrastructure
- Eclipse Update Manager (p2)
- ECF
- Team SVN
- Jabsorb

Thirdparty bundles that we can make use of

- Subclipse from Tigris.org
- P4WSAD from Perforce.com
- RSS OWL from rssowl.org
- SVN client libraries from Polarion

© 2008 Cloudsmith Inc - Made available under the Eclipse Public License.

API QUALITY

- In active use
 - Buckminster is used by a number of companies such as Iona, BEA, Chello, Cape Clear, ?
 - Used by Eclipse Technology project 'Spaces'
 - The Buckminster project uses Buckminster of course, to build and publish Buckminster
- Community participation
 - Users and people who try things out can communicate on the buckminster-dev mailing-list or the buckminster newsgroup. The mailing-list is also mirrored as a newsgroup.
 - We participate in a number of other electronic forums such as pde-build, equinox-dev, cvs-dev, ecf-dev, and subversive newsgroup where we collaborate with the Eclipse community as a whole.
 - We attend build related workshops and all major conferences.
- Documentation
 - Documentation for Buckminster can be found at http://wiki.eclipse.org/index.php/Buckminster_Project

IP CLEARANCE AND LICENSE

- IPZilla Approvals
 - All code for Buckminster resides in the SVN repository at Eclipse.org, it is licensed under EPL, and has been IP approved. We do not include any third party or external projects.
 - See IP Log at: http://www.eclipse.org/buckminster/ip_log.html

SCHEDULE

- Development adhered to Ganymede schedule
 - All milestone deliveries were met

COMITTER DIVERSITY

- Currently 7 committers
 - Inactive committers replaced and one added (since Europa review).
- Committers from multiple countries
 - US
 - Sweden
 - Czech Republic
 - Germany
- Working with / supporting projects
 - Equinox
 - “Ganymatic”
 - Spaces
 - STP
 - ECF
 - EPP

SUGGESTIONS FOR NEXT STEPS

- P2 and Buckminster
 - As functional capabilities in P2 increases, we can gradually replace parts of implementation with P2, as well as making Buckminster functionality available via P2 interfaces.
- Packaging + Buckminster
 - Installer contribution should be common technology
- Maven + Buckminster
 - *Monitor the activity in the new IAM and M2E projects closely and as a solution matures and reaches a valid IP-approval, migrate to make use of that rather than maintaining our own Maven integration.*

VISIT US AT
<http://www.eclipse.org/buckminster>