



Buckminster

Ganymede 2008

<http://www.eclipse.org/buckminster>

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



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.