

# Adobe Chooses Eclipse for Key Product Component

---



For over 20 years, Adobe® has been synonymous with the creative process. When small teams of creative professionals such as graphic designers, photographers and web developers go looking for the digital tools of their trade, they turn to software like Adobe Creative Suite 2. A key component of Adobe Creative Suite 2 is Version Cue®, an embedded client/server toolset that provides workgroups of up to 25 with project management functionality such as version control, file collaboration, and streamlined reviews.

Through continual innovation and development, Version Cue has served the needs of users well for years. In 2004 Adobe undertook changes to the code-base to take advantage of new technologies such as XML that would make the product easier to deploy and maintain.

Selecting Eclipse as the platform for the Version Cue's new architecture was an easy decision for the Version Cue team. Members of the Version Cue development team already had expertise in Eclipse, and they liked what they saw. The multi-platform nature of Eclipse meant that they could easily maintain the server on both OS

“New technology framework initiatives come and go, but the commitment of key players to Eclipse, as well as the enthusiasm of the developer community told us that it is here to stay.”



X and Windows. The swelling user-base and buzz of activity around Eclipse were also reassuring. As Boris Pruessmann, Senior Computer Scientist for Version Cue puts it, “new technology framework initiatives come and go, but the commitment of key players to Eclipse, as well as the enthusiasm of the developer community told us that it is here to stay”.

Building on the Equinox Framework

Of course there was more to choosing Eclipse than just momentum. Eclipse offered Adobe the right technologies for the job. Equinox, the Eclipse implementation of the Open Services Gateway initiative (OSGi) specification, was complete and robust, allowing Adobe to rearchitect the Version Cue server using the Eclipse component model.

For instance, they found that reusing Adobe Creative Suite 2 components in an Equinox environment was as simple as writing a thin compatibility layer.

Developing plug-ins to deliver new functionality in the past had presented significant challenges. Since it was difficult to validate dependencies, Version Cue had not been able to make the most efficient use of system resources by loading exactly what

was needed. Using Equinox and the Eclipse plug-in model, resolving plug-in dependency is simple, and they have been able to achieve true dynamic loading and unloading of plug-ins on demand. The result is a smarter, leaner and faster application that provides a better user experience.

### Efficient Development

Adobe's Version Cue developers saw immediate gains in productivity, and new server functionality

is easier both to write and to maintain. For example, the tools provided by Eclipse for its Plug-in Development Environment (PDE) dramatically reduce development time.

As Pruessmann explains, "It used to take a developer at least 30 minutes to define the initial set-up for a new plug-in. With Eclipse, they achieve the same results in two minutes." Even a new developer on the Version Cue team, using the

Manifest Editor, can quickly create new extension points, and classes are auto-generated for them. Since adopting Eclipse and Equinox, developers no longer break

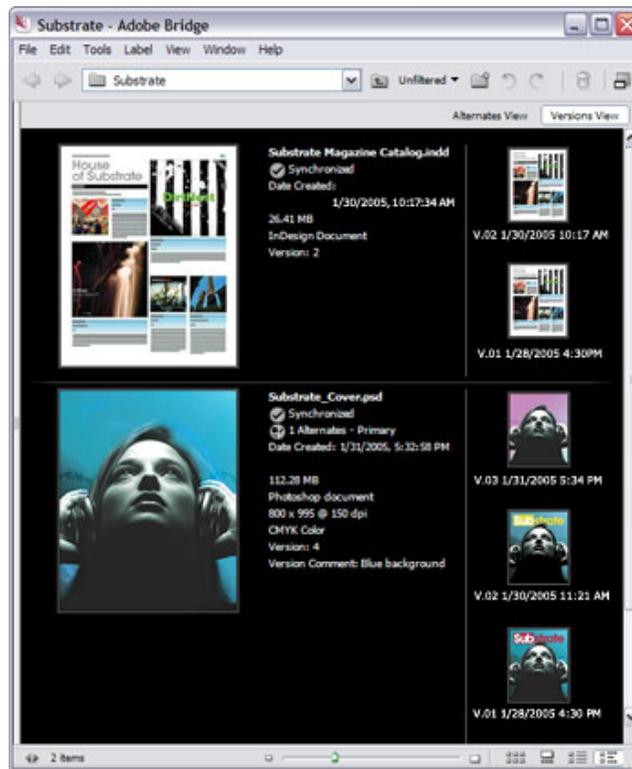
builds, another confirmation that they have made the right choice.

### Achieving the Unexpected

A key measure of any good design decision is the ability to meet new and unexpected needs. The Version Cue team was even able to achieve results that the original designers had never imagined. For example, they already had a standard Adobe loader that they were not at liberty to replace. Although the Equinox loader interface is an area of

Eclipse that developers seldom explore, and is not heavily documented, they were able to integrate their own loader without difficulty. As Pruessmann relates, access to the source code made the task straightforward.

Overall, maintaining Version Cue and developing new features have both been greatly simplified, and the team is convinced that moving to Eclipse was the right choice.



**Adobe has been able to improve development cycles for Version Cue and provide a more responsive client-server experience for its customers.**

*Ron Stone is a technology writer and content management consultant based in Ottawa, Canada.*