

BEA Uses Equinox to Revolutionize Event-Driven Application Development



Event Processing is a new application design pattern focusing on the real time processing of high volume events. BEA customers have embraced this new computing paradigm for systems monitoring, to improve operational visibility and efficiency, for financial front office applications like arbitrage trading, and more.

Event driven applications have extreme performance requirements and require support for event streaming, real time scheduling, real time synchronization and other capabilities specific to event processing that existing application infrastructures cannot support. So BEA looked to a new approach to meet these needs.

Smaller, Lighter, Faster

WebLogic® Event Server 2.0, launched in July 2007, was designed from the ground up to allow the most light-weight, low latency server environment possible. As Robin J. Smith, WebLogic Time and Event Driven Product Manager relates, “We saw the problems that folks in sectors like finance and the

“BEA needs to deliver software of enterprise quality. Equinox provides the robustness we need to make this happen.”



military were having with existing platforms and designed Event Server from the ground up, leveraging our microServices Architecture, based on the Equinox OSGi™ implementation, to give them the leanest, fastest environments possible for mission critical real-time applications.”

When BEA’s developers were creating the Event Server, they considered a number of options, such as using servlets and various Enterprise Java platforms. In the end, they settled on the Eclipse Equinox implementation of the OSGi standard.

Equinox delivered a number of benefits. For example, the architecture allows cleanly modularized components to be isolated at run-time, allowing components to be installed or redeployed on a running application and greatly reducing the need for scheduled downtime. In addition, the advanced class loading feature of Equinox greatly reduced the potential for jar file dependency conflicts at runtime. These features allow for better versioning of product features.

But just as important was the fact that, with mSA, BEA has a technical strategy to host a number of their products on Equinox's OSGi implementation. This allowed the Event Server team to reuse components from other BEA products. For instance, they reused a thread management component from WebLogic Server. They also took advantage of the availability of other open source components such as a Logging Service, and the Spring framework. All of these factors came together to make Equinox the right choice.

Business Focus

The component architecture of the BEA Event Server allows BEA customers to customize and extend their installation to meet their specific business needs. Users create event applications using a tool based on the Eclipse Plug-in Environment (PDE) and the Web Tools Platform (WTP). Since most users are already familiar with extending Eclipse bundles, the learning curve to author Event Server applications was greatly reduced. Another advantage of the Equinox approach was that, unlike developing Enterprise Java Beans (EJB) that require developers to understand detailed technical specifications, Equinox extension points tend to focus on business logic, so users can focus on the business domain, rather than learning a new technical specification.

BEA also used the Spring Dynamic Modules to make it easy to assemble the components that make up an Event Server application. Scott Lewis, WebLogic WorkShop Senior Engineer relates that they have built support for Spring Context Editing directly into their tooling and even extended the standard Spring XML Schema to make it easier to assemble applications.

Equinox's component architecture also allows BEA to organize the Event Server implementation so that customers only have to install the pieces they need. This is a critical advantage for keeping server size down and speed up.

More Efficient Development

The BEA development team found it easier to develop and maintain their code because Equinox allows for well-defined component interfaces. They were able to limit their public APIs to only what customers would need to develop applications. These well-defined APIs make it easier to maintain the code and greatly reduce the effort for integration testing.

The developers also benefited from Equinox's extension mechanism. Instead of writing their own mechanism to allow others to extend the product, BEA was able to reuse the existing Equinox capability. This out-of-the-box solution provides them with a well known and standard method for product extensions.

Managing Dependencies

BEA did run into some issues getting up to speed with Equinox. For instance, establishing and managing hundreds of dependencies between components could be challenging, but Spring Dynamic Modules helped them manage these relationships between bundles. Correctly defining manifest files for components was another challenge that they tackled over time by using better tooling.

Real-time Performance

Customers use Event Server to create applications such as real-time market analysis tools for processing large volumes of data as it comes over the

wire. In many cases, they are able to deliver latencies measured in microseconds to such mission critical applications. The component architecture approach using Equinox was central to making this possible.

One of the earliest markets for WebLogic Event Server was the financial industry, although it has quickly spread to the online gaming, public sector, airlines and other verticals where real-time response is critical. Additional industries that have less demanding response requirements yet still derive significant benefits from event stream processing, such as telecommunications, manufacturing, transportation logistics, and utilities, are now also adopting this technology. .

BEA has big plans for Event Server in the future, and will leverage the Equinox/OSGi model, allowing 3rd party partners to extend the platform with additional solution components.

Alex Alves, WebLogic Time and Event Driven Product Architect knows they made the right decision by choosing Equinox. “BEA needs to deliver software of enterprise quality. Equinox provides the robustness we need to make this happen.”

BEA WebLogic Event Server -- based on BEA's microService Architecture (mSA)

The flexibility and agility of the BEA microService Architecture (mSA) allows BEA to reduce the time-to-market and cost of innovative products like WebLogic Event Server. BEA's mSA-based products are built using the following principles: separation of concerns, modularity and service-based,

as opposed to point-to-point, integration.

At the core of mSA is the Equinox OSGi backplane. BEA uses OSGi technology to build agile component-based solutions that re-use BEA's pre-existing rock-solid technology. In addition, OSGi allows BEA to integrate the best of third-party and open source components to complement its proprietary technology. Because the products are built from exactly the modules needed to provide product functionality, disk and memory footprint are kept as small as possible. BEA has been a leader in bringing the benefits of OSGi technology to enterprise solutions. BEA is a member of the Board of Directors of the OSGi Alliance and a key contributor to its Enterprise Expert Group.

[Ron Stone](#) is a technology writer and content management consultant based in Ottawa, Canada.

March, 2008