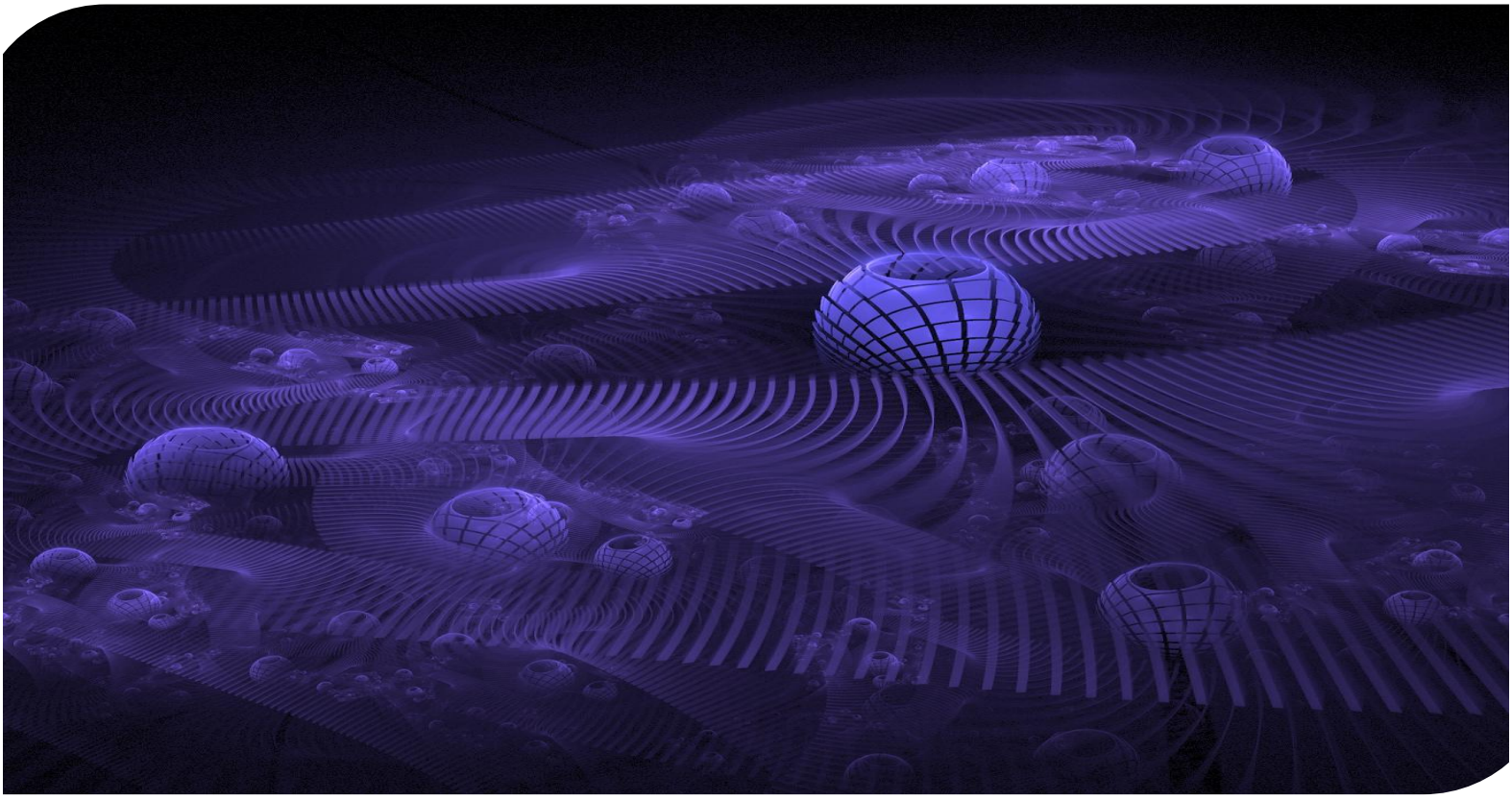
White Paper: Why Upgrade from WebSphere Application Server (WAS) v7 to v8.x?



By TxMQ Publishing Services.
1430B Millersport Highway
Williamsville, NY 14221
+1 (716) 636-0070
TxMQ.com
consulting@txmq.com

One of the more common questions we field at TxMQ comes from the enterprise community. Customers ask: We already upgraded our WebSphere Application Server (WAS) from 6 to 7, why should we now upgrade from 7 to 8? With the amount of chatter surrounding this topic, there's clearly a bit of disconnect, so here's some insight to help in the decision-making process.

There are several compelling reasons to upgrade from WAS v7 to v8, and they center on performance and setup/configuration improvements. The performance gains help you maximize your hardware investments, because you won't outgrow your servers as quickly. That ultimately leads to a reduction in your Total Cost of Ownership (TCO).

The setup/configuration improvements will speed up your end-to-end development cycle. You'll therefore enable better, faster development using the same resources.

Lastly, the mobile-application feature pack offered in WAS v8 is a big advantage for companies already involved with, or wanting to become involved with mobile-app development and operation. This feature pack helps immensely.

That's the broad-stroke look at what a WAS v8 upgrade delivers. Following is a more granular look at the specific features and benefits of a WAS v8 upgrade, including features exclusive to the latest v8.5 update.

Application Server Performance

An upgrade from WAS v7 to v8.x delivers:

- Application performance improvements of up to 20%
- Up to 20% faster server startup time for developers
- Up to 28% faster application deployments in a large topology
  JPA 2.0 optimizations with DynaCache and JPA Level 2 cache

APPLICATION SERVER SETUP

The new Liberty Profile option is a highly composable, fast-to-start and ultra-lightweight profile of the application server and is optimized for developer productivity and web-application deployment.

- Up to 15% faster product installations
- Up to 323% faster application-server creation in a large topology
- Up to 45% faster application-server cluster creation in a large topology
- Up to 11% better vertical scaling on larger multicore systems

JAVA 6

WAS v8 includes JVM runtime enhancements and JIT optimizations. It lowers risks through end-to-end security-hardening enhancements including security updates defined in the Java EE 6 specifications, additional security features enabled by default and improved security-configuration reporting.

SECURITY

Security default behavior is enhanced: SSL communication for RMI/IIOP, protected contents of HTTP session objects, Cookie protection via HttpOnly attribute is enabled.

- Java Servlet 3.0 security now provides three methods – login(), logout() and authenticate() – that can be used with an HTTPServletRequest object and the ability to declare security constraints via annotations
- Basic security support for the EJB embeddable container
- Support for Java Authentication SPI for containers (JASPI)
- Web Services Security API (WSS API) and WS-Trust support in JAX-WS to enable users building single sign on Web services-based applications
- Security enhancement for JAX-RS 1.1

STANDARDS & APIs

The focus on simplification continues in EJB 3.1 with several significant new functions including optional Business Interfaces, Singleton EJBs and Asynchronous Session Bean method invocation.

- CDI 1.0 – New API to support Context and Dependency Injection
- Bean Validation 1.0 – New API for validating POJO Beans
- JSF 2.0 – Adds Facelets as a view technology targeted at JSF
- Java Servlet 3.0 – Makes extensive use of annotations, introduces web fragments and a new asynchronous protocol for long-running requests
- JPA 2.0 – Has improved mapping support to handle embedded collections and ordered lists, and adds the Criteria API
- JAX-RS 1.1 – Delivers Web 2.0 programming model support within Java EE
- JAX-WS 2.2 – Extends the functionality provided by the JAX-WS 2.1 specification with new capabilities. The most significant new capability is support for the Web Services Addressing (WS-Addressing) Metadata specification in the API
- JSR-109 1.3 – Adds support for singleton session beans as endpoints, as well as for CDI in JAX-WS handlers and endpoints, and for global, application and module-naming contexts
- JAXB 2.2 – Offers improved performance through marshalling optimizations enabled by default

NEW & ENHANCED FEATURES

The listings here are important and many.

- The Web 2.0 feature pack – new revenue opportunities and rich user experiences enabled by extending enterprise applications to mobile devices
- Faster migrations with less risk of downtime through improved automation and tools, including a no-charge Migration Toolkit for migrating version-to-version and from competition
- Improved developer and administrator productivity through new and improved features such as improved developer productivity via monitored directory install, uninstall, and update of Java EE applications to accelerate the edit-compile-debug development lifecycle
- Enhanced administrator productivity through automated cloning of nodes within clusters, simpler ability to centrally expand topologies and to locate and edit configuration properties
- Faster problem determination through new binary log and trace framework
- Simpler and faster product install & maintenance with new automated prerequisite and interdependency checking across distributed and z/OS environments
- Deliver better, faster user experiences by aligning programming model strengths with project needs through WebSphere's leadership in the breadth of programming models supported: Java EE, OSGi Applications, Web 2.0 & Mobile, Java Batch, XML, SCA (Service Component Architecture), CEA (Communications Enabled Apps), SIP (Session Initiation Protocol) & Dynamic Scripting
- Integration of WebSphere Application Server v7 feature packs to simplify access to new programming models
- Deliver single sign-on (SSO) applications faster through new and improved support for SAML, WS Trust and WSS API specifications
- Most v7 feature packs are integrated into v8 OSGi Applications, Service Component Architecture (SCA), a Java Batch Container, Communications Enabled Applications (CEA) programming model, and XML programming model improvements XSLT 2.0, XPath 2.0 and xQuery 1.0
- Automatic Node Recovery and Restart

Enhancements in v8.5

The following enhancements are specific to WAS v8.5.

- Application Edition Management enables interruption-free application rollout. Applications can be upgraded without incurring outages to your end-users
- Health Management monitors the status of your application servers and is able to sense and respond to problem areas before end-users suffer an outage. Problem areas include increased time spent on garbage collection, excessive request timeouts, excessive response time, excessive memory and much more
- Intelligent Routing improves business results by ensuring priority is given to business-critical applications. Requests are prioritized and routed based upon administrator-defined rules

- Dynamic Clustering can dynamically provision and start or stop new instances of application server Java Virtual Machines (JVM) based on workload demands. It provides the ability to meet Service Level Agreements when multiple applications compete for resources
- Enterprise Batch Workload support leverages your existing Java online transaction processing (OLTP) infrastructure to support new Java batch workloads. Java batch applications can be executed across multiple Java Enterprise Edition (Java EE) environments
- IBM WebSphere SDK Java Technology Edition v7.0 as an optional pluggable Java Development Kit (JDK)
- Web 2.0 and Mobile Toolkit provides an evolution of the previous feature pack
- Updated WebSphere Tools bundles provide the "right-fit" tools environment to meet varied development and application needs

Are you evaluating a WAS upgrade? TxMQ can help. To get started, contact vice president Miles Roty at (716) 636-0070 x228, miles@txmq.com.

Photo courtesy of Flickr contributor Lynn (https://flic.kr/p/548mSZ)