

An Oracle White Paper  
May 2010

# Learning Oracle GlassFish Server for Tomcat Users

## Introduction

There is a direct connection between the Web container technology used by developers and the performance and agility of applications. This white paper demonstrates the similarities and differences between Oracle GlassFish Server and Apache Tomcat, allowing Tomcat users to make an informed decision about which application server is right for their environment.

For Web 2.0 startups, application service providers, and independent software vendors (ISVs) offering dynamic Web-based applications, the application is the business—and the Web container technology used in the application server can affect business results. The Web container is the part of an application server that manages servlets, JavaServer Pages (JSPs), and other Web-tier components. Web container technology plays a vital role in determining the performance and adaptability of Web-tier software components.

Oracle asserts that the Web container features of Oracle GlassFish Server are far more extensible, modular, easy to use, and adaptable than those of Tomcat, while offering comparable performance. With support for clustering, persistence, messaging, and other key features that Tomcat lacks, as well as enterprise-class service and support from Oracle and the backing of a large and thriving community, Oracle GlassFish Server is a superior alternative for those who wish to move beyond Web-tier application development.

## Overview: GlassFish and Tomcat

Oracle GlassFish Server is the commercial offering based on GlassFish Server Open Source Edition. GlassFish Server Open Source Edition is the open source application server delivered by the GlassFish community, which was originally launched by Sun in 2005. Since its inception, Oracle GlassFish Server has notched 24 million downloads and 300,000 product registrations in a recent one-year span. The current version, Oracle GlassFish Server 3, features an architecture that is modular by default; allows for rapid, iterative Web development; and supports dynamic languages. It is an excellent platform for deploying rich internet applications backed by Java.

Oracle GlassFish Server 2 is a commercially supported Java Platform, Enterprise Edition 5 (Java EE 5)–compatible offering. The commercial version adds an enterprise-quality application server to the options available to the open source community and builds on the foundation laid by earlier versions of Oracle GlassFish Server.

Tomcat was started at Apache by a group that included Sun Microsystems (now part of Oracle Corporation); the initial code drop came from Sun. It became the reference implementation (RI) for early versions of the Java servlets and JSP specifications. The RI for the latest specifications is Oracle GlassFish Server. Tomcat was critical to the early adoption of server-side Java, was available under an open source license, and contributed to the popularity of open source software within enterprise organizations.

## Oracle GlassFish Server: More than a Web Container

In general, applications that run on Tomcat will also run unchanged on Oracle GlassFish Server. However, there are important differences between Tomcat and Oracle GlassFish Server that affect application performance and scalability, usability, and developer productivity.

It is important to note that while Oracle GlassFish Server is a collection of Java EE containers, one of which is a Web container, Tomcat is just a Web container. This crucial difference creates some key advantages for using Oracle GlassFish Server, including the following:

- **Easier migration path.** With Oracle GlassFish Server 2 and Oracle GlassFish 3, there is a clear, straightforward way to take advantage of technologies such as Enterprise JavaBeans (EJB), the Java Persistence API (JPA), Java Message Service (JMS), and other capabilities. With Tomcat, these technologies must be added piecemeal, one at a time. It is the developer's responsibility to implement the capabilities and make sure it all works—and works together.
- **Out-of-the-box support for clustering and failover.** Oracle GlassFish Server 2 can provide clustering and sophisticated high-availability (HA) capabilities enabling applications to meet stringent, enterprise-class service-level agreements (SLAs). Oracle GlassFish Server 3 can support clustering via a load balancer but does not yet have a clustering profile.
- **Superior administration/monitoring.** Oracle GlassFish Server 2 and Oracle GlassFish Server 3 both support centralized administration through a management console and via a command-line

interface (CLI). The callflow monitoring feature in Oracle GlassFish Server 2 enables an application developer or server administrator to determine where an application is spending most of its time. This feature will also be available in Oracle GlassFish Server 3. In addition, third-party vendors can make their software available via the update center feature in Oracle GlassFish Server for easy setup on Oracle GlassFish Server. With Tomcat, new software must be configured in a piecemeal fashion. The update center feature also provides early access to newer versions of technologies such as EJB 3.1, which allows for bundling EJB as a Web Application Archive (WAR) file, rather than having to package the application as an Enterprise Archive (EAR) file.

- **Support for scripting languages.** Oracle GlassFish Server enables running applications written in Ruby/JRuby, Python/Jython, Groovy, PHP, JavaScript/Phobos, Scala, and other languages.

## Comparing Web Container Capabilities

In addition to the overarching advantages of Oracle GlassFish Server as compared with Tomcat, there are additional differentiators between the Web container capabilities of Oracle GlassFish Server versus Tomcat. The lists below summarize a few examples of the advantages.

### Usability/Productivity

Oracle GlassFish Server is more usable and generates more productivity than Tomcat for the following reasons:

- The update center feature (available in Oracle GlassFish Server 2 and Oracle GlassFish Server 3) offers developers point-and-click access to additional features that can extend Oracle GlassFish Server. Examples include Jersey (JAX-RS 1.0) for building RESTful Web services.
- The ability to retain sessions across application redeployments, available in Oracle GlassFish Server 3, is a time-saver for developers creating Java Web applications.
- Alternate docroots at the virtual server and Web applications means that Web applications no longer need to bundle resources as part of the Web application docroot; they may be shared by other Web applications. For example, common images such as icons no longer need to be packaged together, saving time and effort.

### Reliability and Availability

Oracle GlassFish Server is more reliable and available than Tomcat for the following reasons:

- Oracle GlassFish Server 2 and Oracle GlassFish Server 3 provide support for dynamic reconfiguration of virtual servers and HyperText Transfer Protocol (HTTP) listeners, with no server restart required. With Tomcat, if you make a change to a resource pool, you typically have to restart the application server.
- Oracle GlassFish Server supports fine-grained associations between virtual servers and HTTP listeners. A virtual server will receive only those requests that were received on any of the HTTP

listeners that were associated with it. This is extremely useful for virtual hosting, where one could run multiple Websites on the same physical server machine. This feature also helps with better hardware resource utilization, as compared to running each site on a separate machine.

- Oracle GlassFish Server 3 provides superior Comet/Ajax Push to that of Tomcat. With Tomcat, the developer needs to write a lot of code to get the same Comet functionality provided by Oracle GlassFish Server.
- Oracle GlassFish Server 2 centrally manages HA cluster and load balancer agent configuration.

## Security

Oracle GlassFish Server is more secure than Tomcat for the following reasons:

- With virtual server security realms (available in Oracle GlassFish Server 2 and Oracle GlassFish Server 3), you can specify security in many realms, including a file realm, a database realm, and a Lightweight Directory Access Protocol (LDAP) realm. Security realms are specified in the domain's configuration and referenced by Web applications with authentication requirements deployed to the domain. It is also possible to reference security realms at the virtual server level, in which case the security realm applies to all Web applications deployed to the virtual server (a Web application may still override the security realm at the virtual server level with its own). This allows a virtual hosting company to enforce homogeneous security requirements for all the Web applications hosted by it.
- Oracle GlassFish Server supports Java Authentication and Authorization Java Specification Requests (JSRs) 115 and 196.
- Various 118N enhancements in Oracle GlassFish Server 2 and Oracle GlassFish Server 3 allow for HTTP request encoding detection (configurable in sun-web.xml).

## Performance and Scalability

Oracle GlassFish Server is more scalable and offers higher performance than Tomcat for the following reasons:

- Default Web module support in Oracle GlassFish Server 3 improves performance and reduces startup times, compared with previous versions of Oracle GlassFish Server.
- The high-performance, highly scalable Grizzly connector in Oracle GlassFish Server 2 and Oracle GlassFish Server 3 improves request/response throughput. The lower layers of the Web tier in Oracle GlassFish Server are implemented through the Grizzly<sup>1</sup> Framework. The framework is written in Java, taking advantage of the new I/O (NIO) APIs (scalable network and file I/O, which is scalable and also highly customizable).

---

<sup>1</sup>grizzly.dev.java.net

- Oracle GlassFish Server 2 and Oracle GlassFish Server 3 include various performance optimizations, such as “flattened valve invocation,” a modification to the valve architecture that streamlines how each valve gets called, resulting in smaller stack depth and better performance. Oracle GlassFish Server 3 can support Tomcat-style valves as well.
- Oracle GlassFish Server now supports JSR 199, the Java Compiler API, for JSP compilations. Preliminary performance measurements show an order of magnitude improvement in raw javac compilation speed, and an overall execution 3.5 times improved when running JSP Technology Compatibility Kit (TCK) tests.

Extensive scalability testing has been performed in-house to compare the NIO connectors of Tomcat and Oracle Glassfish Server. These tests use a simple servlet to minimize the time spent in the container and measure how many operations per second the various containers can support for increasing numbers of users. For example, at 16,000 users, the in-house benchmark gives the results shown in Table 1.<sup>2</sup>

**TABLE 1. WEB CONTAINER PERFORMANCE AT 16,000 USERS**

	ORACLE GLASSFISH SERVER	TOMCAT
Operations per second	6,988.9	6,615.3
Average response time (ms)	0.242	0.358
Maximum response time (ms)	1.519	3.693
90% response time (ms)	0.6	0.75

## Web Container Feature Alignment: GlassFish and Tomcat

Compounding the advantages of Oracle GlassFish Server and its Web container over Tomcat, the GlassFish community has provided support for well-known features of Tomcat within Oracle GlassFish Server. Table 2 summarizes the Tomcat features that are provided by Oracle GlassFish Server and the additional Oracle GlassFish Server capabilities not currently provided by Tomcat.

<sup>2</sup> [weblogs.java.net/blog/sdo/archive/2007/05/how\\_to\\_test\\_con.html](http://weblogs.java.net/blog/sdo/archive/2007/05/how_to_test_con.html).

TABLE 2. WEB CONTAINER PRODUCT FEATURES

FEATURE	TOMCAT	ORACLE GLASSFISH SERVER 2	ORACLE GLASSFISH SERVER 3
Tomcat valves	x	x	x
mod_jk support	x	x	x
Web-based Distributed Authoring and Versioning (WebDAV) support	x	x	x
Computer-generated imagery (CGI) support	x	x	x
Tomcat bug fixes	x	x	x
Java EE–certified		x	x
Servlet 2.5, JSP 2.1	x	x	x
JavaServer Faces (JSF) 1.2		x	x
Metro Web Services stack (.NET interoperability)		x	x
jMaki (Ajax framework)		x	x
Jersey/JAX-RS 1.0 (RESTful Web services)		x	x
Sophisticated admin console and command-line interface (CLI)		x	x
Java Management Extensions (JMX)		x	x
Node agent and cluster management		x	
Upgrade tool		x	
Application client container		x	
Java Webstart support		x	
JDBC connection pooling	x	x	x
Java Transaction API (JTA)/ Java Transaction Service (JTS)		x	x
J2EE Connector Architecture 1.5		x	x
Native jRuby/Rails deployment—no need for servlet container			x

Dynamic resizing of jRuby Runtime Pool			x
Rapid redeployment—maintain session state on redeploy			x
Update Center feature		x	x
Verification tools		x	
Image Packaging System (IPS) tools			x
Grizzly (Java NIO)		x	x
Comet (HTTP Push)	x	x	x
Common Object Request Broker Architecture (CORBA)		x	
Modular architecture based on OSGi			x
Embeddable server			x
Centralized admin of load balancer plug-in		x	
High availability		x	
In-memory compilation of JSPs		x	x
Integration with Oracle Waveset		x	
Superior admin support for clustering and load balancing		x	



## Conclusion

At all levels, Oracle GlassFish Server is simply a better option than Tomcat for developers and companies aiming to expand beyond Web-tier applications. Benefits of Oracle GlassFish Server include the following:

- Oracle GlassFish Server is a collection of Java EE containers, not just a Web container
- Oracle GlassFish Server is much more extensible and modular than Tomcat, and can be easily expanded as developers take advantage of capabilities such as messaging, clustering, failover, and session management, and use more features such as EJB, JMS, JPA, and Web services.
- Oracle GlassFish Server supports many powerful Web container capabilities that Tomcat does not, and supports many of the best-known Tomcat features.

The choice of an application server is highly strategic for businesses, and Oracle GlassFish Server has proven to be a superior strategic option for developers of next-generation applications.

## For More Information

For additional details about Oracle GlassFish Server, visit [oracle.com/goto/glassfish](http://oracle.com/goto/glassfish). Click “Downloads” to download Oracle GlassFish Server. To learn more about Java EE 6, visit [java.sun.com/javae/](http://java.sun.com/javae/). To join the GlassFish community, visit [glassfish.dev.java.net/public/devindex.html](http://glassfish.dev.java.net/public/devindex.html).



Learning Oracle GlassFish Server  
for Tomcat Users  
May 2010

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2008, 2010, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0510

**SOFTWARE. HARDWARE. COMPLETE.**