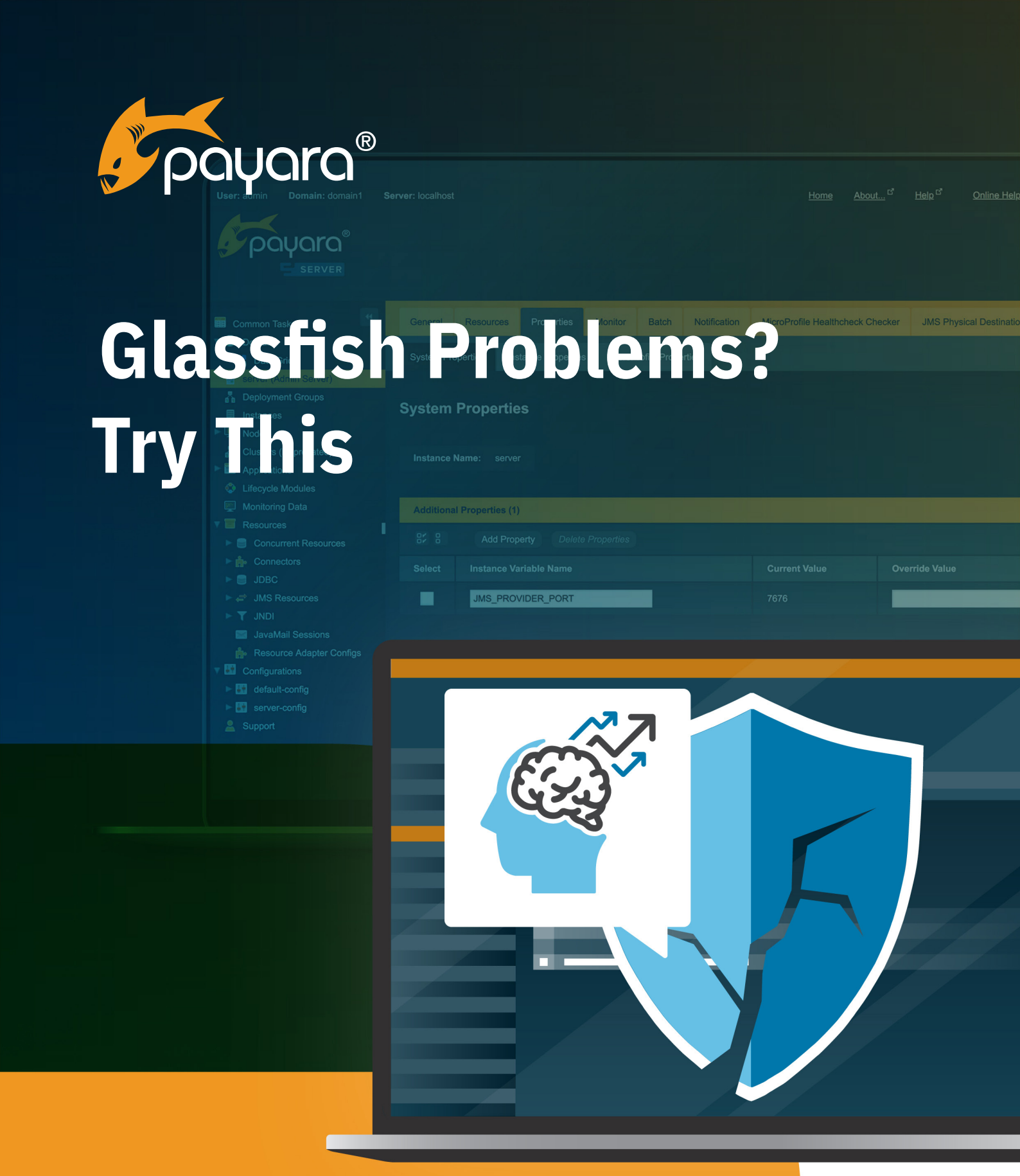
Glassfish Problems? Try This



Power Up Your Jakarta EE

User Guide

Contents

Guide Updated: **October 2024**

The GlassFish application server, once the standard implementation for Java EE, played an important role in deploying production Java EE applications. However, Oracle's decision to discontinue commercial support marked a significant turning point for GlassFish as a viable product for businesses. With the transition of Java EE to the Eclipse Foundation, GlassFish evolved into a community-led, compatible implementation of Jakarta EE.

While GlassFish versions 4.x and earlier continue to operate in many production environments, they face increasing challenges in today's rapidly evolving technological landscape. This document addresses common issues encountered with legacy GlassFish deployments and introduces Payara Server as a modern, supported alternative that offers solutions to these challenges.

Common Challenges with Legacy GlassFish Deployments

1. Limited Automation Support for Server Configuration

GlassFish 4.x and earlier versions lack built-in tools for automating server configuration, making it difficult to:

- Migrate domains and instances between environments
- Provision new servers quickly and consistently
- Integrate with modern DevOps practices and tools

2. Absence of Integrated Real-Time Monitoring

While GlassFish includes a JMX server for gathering metrics, it lacks:

- An internal, user-friendly monitoring system
- Easy access to real-time metrics without external tools
- Simplified remote monitoring capabilities

3. Lack of Official Docker Support

For organizations moving towards containerization, it is important to consider:

- No officially supported Docker images for GlassFish 5.x and later
- Increased complexity in creating and maintaining custom Docker images
- Challenges in leveraging container optimizations introduced in Java 11

4. Limited Server Event Alerts and Notifications

GlassFish doesn't provide:

- Built-in critical server event alerts
- Flexible notification systems for various channels (e.g., email, Slack, monitoring services)
- Easy integration with modern DevOps monitoring stacks

5. Outdated Security Standards Support

As security requirements evolve, legacy GlassFish versions struggle with:

- Limited support for modern authentication mechanisms like OpenID Connect
- Lack of built-in integrations with popular identity providers
- Challenges in implementing current best practices for application security

6. Lack of MicroProfile Support

GlassFish versions 4.x and earlier:

- Have no support for MicroProfile specifications
- Lack essential features for developing cloud-native Java microservices
- Cannot leverage the benefits of standardized APIs for microservices architecture

The absence of MicroProfile support in older GlassFish versions creates significant challenges for organizations looking to modernize their applications and adopt cloud-native architectures.

How Payara Addresses GlassFish Challenges

1. Automation Assistance with Asadmin Recorder

Payara Server introduces the Asadmin Recorder, a powerful feature that:

- Records configuration changes for easy script generation
- Simplifies the creation of automated deployment and configuration processes
- Integrates seamlessly with popular DevOps tools

2. Internal Monitoring Solution

Payara's built-in monitoring console (Community Edition) and Payara InSight (Enterprise Edition) provide:

- Real-time metric browsing without external tools
- Pre-configured metric dashboards for quick analysis
- Customizable monitoring pages for specific organizational needs

3. Supported Docker Images

Payara offers officially supported Docker images that:

- Cover both Full and Web profiles of Java EE
- Are regularly updated and optimized for production use
- Leverage Java 11+ optimizations for containerized environments

4. Comprehensive Alert and Notification System

Payara Server includes:

- A Health Check service for monitoring critical server metrics
- A Request Tracing service for performance analysis
- A flexible Notification service supporting various channels (e.g., email, Slack, Datadog)

5. Modern Security Standards Support

Payara provides built-in support for:

- OpenID Connect via annotations for easy integration
- OAuth2 support for modern authentication flows
- MicroProfile JWT for securing JAX-RS services

6. Comprehensive MicroProfile Support

Payara Server provides full support for MicroProfile specifications, offering:

- A complete set of APIs for developing cloud-native Java microservices
- Standardized solutions for configuration, fault tolerance, health checks, metrics, and more
    - Seamless integration of MicroProfile features with Java EE applications

This comprehensive MicroProfile support enables organizations to:

- Modernize existing applications more easily
- Develop new cloud-native services using standardized APIs
- Improve application resilience, observability, and maintainability in distributed environments

GlassFish vs Payara: Problem-Solution Comparison

Area of Concern	GlassFish Problem (v4.x and earlier)	Payara Solution
Server Configuration Automation	Limited built-in automation tools; manual configuration often required	Asadmin Recorder for easy script generation and integration with DevOps tools
Monitoring and Metrics	Relies on external JMX clients; difficult to set up secure remote monitoring	Built-in Monitoring Console (Community) and Payara InSight (Enterprise) for real-time metric browsing and analysis
Containerization Support	Lack of official Docker images for older versions; manual image creation required	Official, supported Docker images for all Payara Enterprise versions.
Event Alerts and Notifications	No built-in alert system; reliance on external monitoring tools	Comprehensive Health Check, Request Tracing, and Notification services with support for various channels (e.g., email, Slack, Datadog)
Security Standards	Limited support for modern authentication mechanisms	Built-in support for OpenID Connect, OAuth2, and MicroProfile JWT
MicroProfile Support	No support for MicroProfile specifications	Full support for corresponding MicroProfile specifications, enhancing cloud-native Java development
Commercial Support	Limited commercial support by smaller ISVs	Full commercial support with regular updates and security patches

Area of Concern	GlassFish Problem (v4.x and earlier)	Payara Solution
DevOps Integration	Limited tools for modern CI/CD pipelines	Improved automation, monitoring, and Docker support for seamless DevOps integration
Performance Optimization	Basic JMX-based monitoring; limited tools for identifying bottlenecks	Advanced monitoring and profiling tools for efficient performance optimization
Future Compatibility	No updates for newer Java and Jakarta EE versions	Active development ensuring compatibility with latest standards and technologies
Total Cost of Ownership	Potentially high due to custom development and maintenance needs	Initial licensing costs offset by reduced custom development and improved operational efficiency

Business Reasons for Transitioning from GlassFish to Payara

Business Reason	Explanation
Reduced Operational Risk	• Payara offers full commercial support, regular updates, and security patches for all Payara Enterprise releases from version 4 to 7 • Minimizes risks associated with running unsupported software in production • Ensures timely resolution of critical issues
Improved DevOps Integration	• Payara's automation tools align better with modern DevOps practices • Official Docker support facilitates containerization efforts • Advanced monitoring capabilities enhance operational visibility

Business Reason	Explanation
Enhanced Security Posture	• Built-in support for current security standards (e.g., OpenID Connect, OAuth2) • Easy integration with popular identity providers • Reduces need for custom security implementations • Helps maintain a stronger overall security stance
Lower Total Cost of Ownership	• Reduction in custom development needs • Easier maintenance due to built-in features and tools • Improved operational efficiency can lead to significant cost savings over time • Potential offset of initial licensing costs through long-term benefits
Futureproofing	• Active development ensures compatibility with latest Java and Jakarta EE standards • Full support for MicroProfile specifications • Enables businesses to stay current without major migration efforts • Facilitates adoption of cloud-native architectures
Performance Optimization	• Advanced monitoring and profiling tools help identify performance bottlenecks • Easier resolution of performance issues • Potential improvement in application responsiveness • More efficient resource utilization
Simplified Compliance	• Regular security updates help meet regulatory requirements • Improved logging and monitoring aid in audit processes • Standardized security features simplify compliance implementations
Accelerated Innovation	• MicroProfile support enables rapid development of cloud-native services • Modern tooling allows faster feature development and deployment • Standardized APIs reduce learning curve for developers
Improved Scalability	• Better support for clustering and distributed architectures • Cloud-friendly features facilitate scaling in dynamic environments • Improved resource management for handling varying loads
Enhanced Developer Productivity	• Comprehensive documentation and community support • Standardized APIs across Java EE and MicroProfile reduce complexity • Built-in tools reduce time spent on boilerplate tasks

Conclusions

While GlassFish served as a reliable application server for many years, the lack of commercial support and the rapid evolution of enterprise Java ecosystems have left many organizations facing significant challenges. Payara Server, as a commercially supported replacement for GlassFish, offers a clear path forward, addressing these challenges while providing additional features that align with modern enterprise requirements.

By transitioning to Payara, businesses can reduce operational risks, improve developer productivity, enhance security, and position themselves to leverage the latest advancements in Jakarta EE and MicroProfile technologies. This transition not only solves immediate issues faced with legacy GlassFish deployments but also provides a foundation for future growth and innovation in enterprise Java applications.

sales@payara.fish

UK: +44 800 538 5490
Intl: +1 888 239 8941

www.payara.fish