GlassFish 7 To Payara Platform Enterprise Migration

Power Up Your Jakarta EE

User Guide

Contents

Guide Updated: **June 2024**

The GlassFish application server, previously the standard implementation for Java EE, was once a reliable platform for deploying production Java EE applications. However, when Oracle discontinued commercial support, it marked a 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.

If your organisation relies on GlassFish for production workloads, it's important to consider the benefits of a commercially supported runtime for your Java and Jakarta EE applications. While a community product might suffice in the short term, the lack of predictable updates, reliable security patches and professional support from a trusted vendor can expose your systems to unnecessary risks and vulnerabilities.

For example, the average costs of security breaches is currently estimated around 4.5 million dollars[1] and it continues to increase. As a stable, fully supported solution, Payara Server Enterprise addresses these concerns by providing the stability, security and expert guidance your business needs to succeed.

With its roots in GlassFish, this solution initially emerged as a drop-in replacement when Oracle ended its support. Over time, Payara Server has grown into a fully supported commercial Java EE (and now Jakarta EE) compatible implementation with a rich feature set that diverges significantly from GlassFish.

Why Payara Platform Enterprise instead of GlassFish?

Production-Ready and Stable with Full Support: Payara Server Enterprise offers a 10-year software lifecycle for any given major version. Users can benefit from monthly bug fixes and patches, ensuring security and stability . You also get fast issue resolution and priority on feature requests directly from our global engineers.

Cloud-Native and Aggressively Compatible: Payara Server Enterprise is optimized for cloud, on-premises and hybrid environments. It's container-friendly, supports popular cloud services, and is ready for future Jakarta EE compliance. In effect, it was one of the first to earn Jakarta EE certification, demonstrating its strong commitment to this standard. In addition, subscription costs are competitive for cloud-users and remain consistent across environments.

Open-Source Software with a Future You Help Define: Payara Server Enterprise is a fork of GlassFish, and our Enterprise customers can directly influence its evolution through customer advisory calls. The open-source community also welcomes your ideas, feedback and collaboration to help advance the solution.

Replaces Shoal: Payara Server Enterprise includes Hazelcast for an embedded Data Grid, providing equivalent functionality to Shoal but easier management and enhanced security (encryption at rest and in transit).

Simplified Deployment with Deployment Groups: Easily add new Payara instances using templates and automatically install software via SSH. Deployment Groups simplify artefact deployment across multiple instances.

Advanced Request Tracing: Go beyond basic endpoint tracing with insights into EJB beans, CDI beans, Servlets and Jakarta REST resources that help you identify performance bottlenecks in complex applications.

Comprehensive JDK Support: Payara Platform Enterprise customers benefit from JDK support by Azul, ensuring expert assistance for JVM-related issues and ongoing access to JDK 8+ security updates.

Cloud-Native and Container-Friendly: Payara Server offers official Docker images, specialised Payara Micro packaging for microservices as well as seamless integration with Kubernetes and Docker Swarm.

Innovation and Community: Payara actively shapes the future of Jakarta EE and MicroProfile, contributing to the development of new specifications and features. You have the flexibility to choose the right mix of technologies for your use case and gradually migrate to microservice architectures.

While smaller entities may provide some support for GlassFish, it's important to remember that it remains a community-driven project. It lacks the comprehensive backing and reliable support infrastructure that a commercial offering like Payara Server Enterprise can provide. For businesses with mission-critical production workloads, relying solely on a community product introduces an element of uncertainty and risk that can be avoided by choosing a platform with dedicated, professional support.

Building on our comprehensive guides for migrating from GlassFish 3 and 4, this new guide focuses on transitioning from GlassFish version 7 to the latest Payara Server Enterprise release. If your

applications are currently running on older GlassFish versions, refer to our previous guides on upgrading to the corresponding Payara Server versions. Here, we'll concentrate on navigating the specific path from GlassFish 7 to Payara Server Enterprise 6.

The Migration Map

Migrating from one runtime to the other can be a daunting task. However, proper planning and execution can help mitigate any potential mishaps that can happen. The following is a broad outline of the suggested migration map from GlassFish 7 to Payara Server Enterprise 6. Each step in the following list is detailed in subsequent sections.

1. **Preparation**
    - Ensure compatibility with JDK 11 or later
    - Review your application's dependencies for compatibility with Jakarta EE 9/10
    - Back up your GlassFish 7 domain.
2. **Migration To Payara Platform Enterprise**
    - Install Payara Platform Enterprise 6
    - Use the restore-domain command to migrate your backed-up GlassFish domain
    - Built-In Database
    - HTTP/2 Protocol Support.
3. **Clustering and High Availability**
4. **GlassFish Server Control Features**
    - Monitoring Scripting Client
    - Performance Tuner
    - Load Balancer Configurator Plugin
5. **Features to Consider During or After Migration**
6. **Known Issues After Migration.**
7. **Testing and Validation**
    - Thoroughly test your migrated applications in a staging environment
    - Monitor performance and resource utilisation.

Preparation

Before starting the migration process, it's important to prepare the base requirements for a smooth migration process.

JDK Compatibility

Ensure your environment is upgraded to at least JDK 11 or a more recent version. Payara Server Enterprise 6 is optimised for these newer JDK releases, offering improved performance and compatibility with the latest Jakarta EE specifications.

Dependency Assessment

Before migrating to Payara Platform Enterprise 6, make sure your application is compatible with the latest Jakarta EE version (currently Jakarta EE 10) and your JDK. We recommend upgrading to the latest Jakarta EE before migrating to Payara Platform Enterprise. If your application is running on Java EE 8 (Jakarta EE 8) or older, refer to our Jakarta EE upgrade guide for instructions.

Both GlassFish 7 and Payara Platform Enterprise 6 support Jakarta EE 10, but if you need to migrate to Payara Platform Enterprise with an earlier Jakarta EE version, please consult our other migration guides for compatible versions of Payara Server Enterprise.

Identify any outdated or incompatible libraries and update them accordingly. This step will help prevent unexpected issues during and after migration.

Domain Backup

Domain backup involves creating a snapshot of your GlassFish domain configuration and data, which can then be restored on Payara Server.

GlassFish Domain Backup

The `asadmin` command line utility in GlassFish supports domain backup and restore. The process typically involves using the `backup-domain` command, specifying the backup directory and domain name as follows:

```
asadmin> backup-domain --backupDir <path-to-backup-directory> <domain-name>
```

This command will create a compressed file containing the domain backup in the specified directory. Once you have prepared your environment and backed up your GlassFish domain, you can proceed with migrating it to Payara Platform Enterprise 6.

Migration To Payara Platform Enterprise

Once you have a base JDK and have backed up your GlassFish domain, it is time to restore the same on Payara Server Enterprise.

Install Payara Platform Enterprise 6

Download the Payara Platform Enterprise 6 distribution from your customer portal. **If you are not yet a Payara customer, you can download a trial copy** here**.**

Restore The GlassFish Domain

Once you have installed Payara Server, you can restore the GlassFish domain backup using the `restore-domain` command as follows:

```
asadmin> restore-domain --filename <path-to-backup-file> --domaindir <domain-directory>
```

Replace <path-to-backup-file> with the actual path to your backup file and <domain-directory> with the desired location for the restored domain on Payara Server.

Important Considerations

- Node and Instance Migration: If your GlassFish domain includes instances on separate nodes, you'll need to install Payara Server on those nodes and re-synchronize the instances with the Domain Administration Server (DAS).
- Configuration Adjustments: After restoring the domain, review and adjust the configuration as needed, especially for data sources and security realms, to ensure compatibility with Payara Server.

Built-In Database

Payara Server Enterprise has replaced the outdated Derby DB with H2. This primarily impacts internal server functions, such as persistent timers and batch job history. While this change should not affect most applications, it's recommended to verify compatibility with H2 coming from GlassFish, if Derby was the default. **Neither Derby nor H2 are suitable for production. We recommend using a dedicated database for these environments.**

HTTP/2 Protocol Support

In addition, Payara Server Enterprise now supports HTTP/2 by default on secure listeners. Due to potential compatibility issues, it's adviseable to disable HTTP/2 during migration and gradually re-enable it for testing after your application is running smoothly on Payara Server Enterprise.

Clustering and High Availability

GlassFish's reliance on the Shoal project for clustering presents numerous challenges in terms of complex setup, manual scaling, limited data replication, and potential performance degradation over time. Payara Server Enterprise skirts these limitations with the Domain Data Grid feature, a dynamic and scalable solution powered by Hazelcast. This grid automatically integrates all instances within a domain, ensuring seamless data sharing and eliminating the need for manual configuration when adding new instances.

Deployment Groups, another innovative feature of Payara Server Enterprise, build upon the Domain Data Grid, providing even more flexibility in cluster management. Unlike GlassFish's rigid clusters, Deployment Groups allow you to add or remove instances dynamically, assign them to multiple groups, and even incorporate Payara Micro instances.

While maintaining backward compatibility with GlassFish cluster configurations, Payara Server encourages migration to Deployment Groups for enhanced functionality. Instances in Payara Server are treated as standalone by default and they automatically join the Data Grid, simplifying administration.

Payara Server Enterprise offers a variety of clustering options: Domain Data Grid, Deployment Group and legacy clusters (deprecated). Understanding these options and choosing the right one to address your specific requirements will help smoothen the migration journey.

Server Control Features

Monitoring Scripting Client

Payara Server offers a comprehensive suite of monitoring tools, surpassing the capabilities of GlassFish's Monitoring Scripting Client.

For real-time visibility into JVM and server statistics, Payara Server provides an embedded JMX server with AMX MBean support, allowing you to plug in compatible consoles, such as JVisualVM or Zulu Mission Control.

Payara Server also supports MicroProfile Metrics, enabling automated, real-time metrics generation within your application code for use by tools like Prometheus. Furthermore, Payara makes this process extremely simple by automatically exposing JMX/AMX MBean properties as vendor-scoped metrics, reducing manual configuration efforts.

In addition, Payara Server offers a Health Check service that monitors essential system resources and notifies you when predefined thresholds are reached. For in-depth monitoring and analysis, Payara Insight, our fully customizable monitoring Web console, provides real-time data aggregation from multiple nodes. The insights asre presented visually in an intuitive manner. This powerful tool is continually evolving to meet the needs of production environments.

Performance Tuner

While GlassFish offers a Performance Tuner tool to automatically adjust settings based on a questionnaire, Payara Server does not currently have an equivalent feature. However, recognizing the need for production-ready configurations, Payara Server Enterprise includes a pre-tuned "production" domain and domain template with optimised settings that cater to typical production environments. This gives users a valuable starting point when migrating from GlassFish.

Load Balancer Configurator Plugin

Payara Server does not currently offer a direct replacement for GlassFish's Loadbalancer Configurator Plugin. However, we recommend setting up a load balancer manually using either Apache Web Server with the mod_jk Tomcat connector. Alternativelym you can use an Nginx web server with sticky sessions enabled through its available plugins. These approaches can drive broader compatibility with various load balancer options and gives you greater control over your configuration.

Features to Consider During or After Migration

There are lots of features added in Payara Server Enterprise that are not available in GlassFish. These are designed to give your application the best of class performance and reliability. They include:

Payara Micro

Payara Micro, a specialized distribution of Payara Server, is specifically engineered for microservices and cloud-native applications. Its compact size and streamlined design make it exceptionally well-suited for containerized environments. Additionally, Payara Micro simplifies the deployment process and offers automatic clustering capabilities, ensuring your Jakarta EE applications can be easily deployed and scaled in modern cloud infrastructures.

Slow SQL Logging

This function help users pinpoint performance bottlenecks in database interactions by logging queries exceeding a specified time threshold.

Payara Health Check Service

Through this feature, users can proactively monitor system resources like CPU, memory and garbage collection to detect potential issues.

Request Tracing Service

Supporting the quick identification of performance bottlenecks, the Request Tracing Service empowers users to trace requests across various components (REST endpoints, EJBs, CDI beans, etc.) to identify performance bottlenecks.

Enhanced Cloud Deployment

Payara Server offers multiple discovery modes, including Kubernetes support, for seamless integration with cloud environments and elastic scaling.

Default Role/Group Mapping

This capability gives users the opportunity to configure default role mappings directly in deployment descriptors for greater portability.

Comprehensive Administration and Monitoring Tools

Users can leverage a full-featured web console, scriptable CLI, RESTful API and JMX MBeans for comprehensive administration and monitoring.

Testing and Validation

After migrating your GlassFish domain to Payara Server Enterprise 6, rigorous testing and validation must be carried out to ensure everything worked as expected. The following are pointers to help ensure all goes well:

- **Staging Environment:** Deploy your migrated applications in a staging environment, or stage, that mirrors your production setup. By doing so, you can easily identify and address any compatibility issues or unexpected behaviour before impacting your live systems.
- **Comprehensive Testing:** Conduct thorough testing of all application functionalities, including core business logic, integrations and user interface tests. Use automated tests, load tests and stress tests to simulate real-world scenarios and validate performance under various conditions.
- **Performance Monitoring:** Closely monitor the performance of your migrated applications on Payara Server Enterprise. Pay attention to metrics like response times, throughput and resource utilisation (CPU, memory, disk I/O). Compare these metrics with your GlassFish baseline to identify any performance regressions or areas for optimization.
- **Resource Use:** Analyse the resource utilisation of your applications on Payara Server. Ensure that the server is adequately provisioned to handle the workload and that resources are allocated efficiently. Fine-tune JVM settings, connection pools and other configuration parameters as needed to optimise performance and resource use.

Summary

As a commercially unsupported runtime, GlassFish can expose current mission-critical applications in production environments to a number of risks. Therefore, businesses using this legacy solution should transition to key alternatives to protect their applications and advance their capabilities.

This guide details the migration process from GlassFish 7 to Payara Server Enterprise 6, highlighting the advantages of switching to a commercially supported Jakarta EE runtime. Payara Server offers numerous benefits over GlassFish, including long-term support, regular updates, cloud-native features and a dedicated engineering team behind it. Thanks to these elements, Payara Server simplifies deployments, provides advanced request tracing and JDK support.

The guide outlines the migration roadmap, addressing potential challenges, such as database changes, HTTP/2 protocol adjustments and clustering considerations, and how to address them. It also introduces new Payara features, like Slow SQL Logging and the Health Check Service.

To ensure a smooth transition, the guide emphasises the importance of testing and validation in a staging environment before deploying to production. If you're looking for a stable, secure and future-proof platform for your Jakarta EE applications, consider migrating to Payara Server Enterprise.

Contact us today for full migration support and take your mission-critical workload to the next level of performance, stability and security with Payara Server Enterprise.

References

1. IBM. (2023). The Cost of a Data Breach Report 2023, Available at: https://www.ibm.com/reports/data-breach [Accessed: 24 June 2024].

sales@payara.fish

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

www.payara.fish