Migrating from TomEE to Payara Server 5

Migration Guide

The Payara® Platform - Production-Ready,
Cloud Native and Aggressively Compatible.

Contents

Introduction

The Apache TomEE product is built on top of Apache Tomcat. Tomcat is a Web server that can be used to create dynamic websites. When you develop an Enterprise Application, you will need to integrate more frameworks than what is needed to create a dynamic website. Since Tomcat is designed for dynamic website creation, frameworks for validations, connections with messaging systems, security, providing REST endpoints, etc. , are not integrated. Tomitribe has started with the Tomcat Web Server as a basis and integrated various Apache projects that implement all specifications of Java EE, now Jakarta EE, to create their TomEE project

The TomEE server implements the Java EE specification and some versions have support for the MicroProfile specifications. Payara Server is based on the Jakarta EE and MicroProfile standards and since the TomEE server only provides a few specific features added on top of the Java EE standards, migrating from TomEE to Payara should be extremely easy to do.

Releases

TomEE releases are irregular and not announced on the blog or the company website. The only information I could find is a tag in the Git repository indicating when a new version should be available. There is a gap of one to 4 months between each release. It is not clear on which basis the decision is made to release an updated version.

This is in contrast with Payara where you have monthly releases of the Payara Platform Enterprise Edition and regular and frequent releases of the Payara Platform Community Edition. A new Payara release can contain new features as well as bug fixes, component upgrades, and security fixes. This to make sure that you can always run the most secure and best version of Payara.

From the TomEE download page, there is the option to choose one of the 2 major versions that are supported. TomEE 8 is an implementation of the Java EE 8 specifications and TomEE 7 targets the Java EE 7 implementation.

Payara Server also supports the same target specifications as TomEE. Payara Server 4, available for our Enterprise customers, is built to run Java EE 7 applications, and Payara Server 5 runs Java EE 8 applications and is a Jakarta EE 8 certified implementation.

If you are running your application on one of these two TomEE releases, the application will run fine on Payara Server as they are created to support the same version of the specification. The migration will be very straightforward, just perform the required configuration on Payara Server and deploy the application.

TomEE Versions

TomEE comes in four flavors TomEE WebProfile, TomEE MicroProfile, TomEE +, and TomEE PluME. TomEE is the basic flavor supporting the Jakarta EE web profile. The PluME version includes all libraries that are required for a Jakarta EE Full Profile and it is not only based on Apache projects but contains some frameworks like EclipseLink to better match the Glassfish stack for easier migrations.

The two other versions are designed for use if your application is based on REST endpoints, use the TomEE MicroProfile version, or if your application is based on SOAP endpoints, use the TomEE + version.

TomEE WebProfile	Jakarta EE web profile
TomEE MicroProfile	Jakarta EE web profile + MicroProfile
TomEE +	Jakarta EE web profile + SOAP support
TomEE PluME	Jakarta EE Full Profile

Payara Server has two versions, the Payara Server Web Profile which corresponds to the basic TomEE version, and the Payara Server Full Profile which contain all Jakarta EE specifications and thus corresponds to the TomEE PluMe version. If you are running one of the two other versions of TomEE, Payara Server Full Profile can run your application without a problem as it includes the same specifications and also has support for MicroProfile.

Payara also has the Payara Micro product if your application is targeted at microservices or cloud environments. It contains the Jakarta EE Web profile specification with the addition of the Concurrency and Batch specifications. Due to this support for running your application through the hollow jar technique, it might be more suited for those environments.

Standards Support

TomEE 8 is not a certified implementation (https://tck.work/tomee/projects) of Java EE 8 or Jakarta EE 8. It passes around 98% of the Jakarta EE 8 tests, failing some tests for the JASPIC module. Only half the modules have 100% coverage. This means that your Jakarta EE application runs fine on TomEE but some functionality will not work as expected as several parts are missing in TomEE, around the security features defined in Jakarta EE.

This is in contrast with Payara Server which is a Jakarta EE 8 compatible product. All tests from the Technology Compatibility Kit (TCK) run successfully on Payara since October 2019 and is verified monthly with each release since then.

The TomEE versions, except the basic TomEE version, also have support for the MicroProfile Specification, but the support is only limited to version 1.3 released at the beginning of 2018. So, the supported specification is rather old and targeted to the Java EE 7 implementations. Since the philosophy of MicroProfile is to provide the user with some innovative, fast-moving functionality when creating cloud-native applications, supporting a 3-year-old version defeats the goal of MicroProfile.

Payara Server on the other hand supports each new version of the MicroProfile specifications a few months after they are officially released. Currently, Payara Server implements the latest version MicroProfile 4.0.

In the latest release of Jakarta EE 9, there is a major change in the package names and a general namespace change to enable the platform to move forward in the open-source space, leaving the Sun and Oracle Java EE legacy behind. TomEE 9 is a preview that implements this new Jakarta Namespace, but this version can run only Jakarta EE 9 applications and not the ‘older’ Jakarta EE 8 and Java EE applications.

The new Jakarta EE 9 namespace is already in technical preview on Payara Server 5 for the moment. But in contrast with TomEE, we do not require you to convert to the new namespace, you can still use the older specifications. Due to the Eclipse Transformer technology, we detect that the user deploys a Jakarta EE 9 application, and we transform this to the older javax namespace on the fly. This allows you to deploy a Jakarta EE 9, a Jakarta EE 8, and a Java EE application on Payara Server 5. It gives users an easy upgrade path at their own pace. We expect to pass the requirements for the Jakarta EE 9 certification in the first half of 2021.

Supported JDKs

Just as with the announcements of the new releases, there is no official statement on TomEE's supported JDK versions. Looking around at some of the examples and conversations on forums, TomEE seems to run fine on JDK 8 and JDK 11.

Payara Server is tested thoroughly on Java LTS versions 8 and 11. We also test Payara Server on different implementations of the JDK, including OpenJDK based implementations, but also Eclipse J9 for example, to make sure you have a smooth experience on any JDK.

Our Payara Enterprise customers also have included support for any JDK issues when running on the Azul Zulu JDK. It guarantees that you get a resolution for any problem whether it is Payara or JDK-related.

Configuration

The configuration of TomEE must be performed manually in several configuration files. Some of the files are used by Tomcat and others by the TomEE extensions. The format is a bit strange as the main layout of the files is XML based but within a tag, you need to use a property-like style. For example.

```
1.    <Resource id="moviesDatabase" type="DataSource">
2.    JdbcDriver com.mysql.jdbc.Driver
3.    JdbcUrl jdbc:mysql:localhost:3306/moviesdb
4.    UserName sa
5.    Password secret
6.    JtaManaged true
7.    </Resource>
```

There is no command-line tool or graphic tool to help with this configuration which makes entering the correct configuration tedious and error-prone in some cases.

Payara Server has the Asadmin CLI that allows reading and writing the configuration through the command line. It verifies all commands and values so that configuration values are correct and do not result in issues at runtime. The usage of the command line tool allows you to perform the configuration in a repeatable, error-prone fashion. If you are not familiar with all the commands and options, you can first use the web-based Admin Console. The Admin Console guides you through configuration of Payara Server. It also has the Asadmin recorder to generate the equivalent Asadmin CLI for the actions you perform in the Payara Server Admin Console, which helps you learn all the commands and configure Payara Server in a scripted way.

Monitoring

TomEE does not contain any specific tools for monitoring. You have the availability to check some basic parameters such as the number of Web Sessions through the Tomcat Web Session Manager. With the help of a JMX client, you can also check the values of the JMX beans as part of your monitoring strategy. TomEE can also be integrated with Application Performance Management (APM) tools like Datadog, newRelic, and AppDynamics.

Payara Server has many built-in monitoring tools and can also be integrated with external APM tools. The Monitoring Service keeps an eye on the core parts of Payara Server and reports about Slow SQL query performance and Stuck Threads to name a few. For each submodule, you can configure the level of detail that is gathered.

The monitoring data can be accessed through a JMX client, exported to a Prometheus instance, or accessed through Payara Insight, our graphic monitoring and alerting tool. But these values can also be fed into the Health Service of Payara that compares the values against some thresholds and sends alert messages through the notification channels when the thresholds are reached. By default, Payara Server has notifications channels for the classic log file, messaging systems like e-mail, SMS, chat (Discord and Microsoft Teams), message queue systems, and external tools.

Serverless

The TomEE documentation mentions that it has support for serverless. Looking more in detail, the topic describes how you can start TomEE in an embedded mode. As part of your application, you can start TomEE and an application so that it can respond to user requests.

Payara Server is also available in an embedded format. It works almost identical as in the TomEE case. The following snippet can be used to start Payara Server and deploy an application.

```
1.  BootstrapProperties bootstrap = new BootstrapProperties();
2.  GlassFishRuntime runtime = GlassFishRuntime.bootstrap(bootstrap);
3.  GlassFishProperties glassfishProperties = new GlassFishProperties();
4.  glassfishProperties.setPort("http-listener", 8083);
5.  glassfishProperties.setPort("https-listener", 8184);
6.  GlassFish glassfish = runtime.newGlassFish(glassfishProperties);
7.  glassfish.start();
8.  File war = new File("/path/to/application.war");
9.  Deployer deployer = glassfish.getDeployer();
10. deployer.deploy(war)
```

If you are migrating from TomEE to Payara Server only a few changes to the application code are required to startup Payara Server in an embedded way instead of TomEE. No drastic adaptation is required.

Clustering

You can cluster multiple TomEE instances to distribute the user load. The TomEE clustering depends entirely on the Tomcat clustering capabilities. It supports only the distribution of the HTTP session information. Requests from the user can be routed to any instance of the cluster as all the session information is available on them.

Payara Server has many more cluster features than the HTTP Session distribution. It contains the concept of the Deployment Group where you can have a centralized configuration of the cluster and application. When performing actions against this Deployment Group, the information is sent to all instances belonging to this Deployment Group, which makes setting up and configuring a cluster as simple as a single instance. The runtime itself makes sure that the cluster is functioning properly with all the settings you have made centrally.

Not only can you share HTTP session information, but you can also define a cluster-wide EJB singleton and store any kind of information in the Domain Data Grid.

Testing Tools

Besides Tomcat, the Apache OpenEJB component is one of the two main components of the TomEE distribution. It implements the EJB specification and is an important part that upgrades Tomcat to an application server with all Enterprise features. The OpenEJB container can be started separately, which is mainly used for testing purposes to test the application functionality that is only dependent on the EJB container.

Payara Server does not have an EJB implementation that can be started separately as the implementation is not composed of separate frameworks and libraries, but instead is composed of tightly integrated modules so that it performs much better.

If you are using the OpenEJB in a standalone way to perform integration testing, the recommended way is to modify the tests to use the Arquillian test framework. This allows you to test your application in a more realistic scenario as you can use Payara Server in an embedded or remote configuration that is almost identical to your production situation. This guarantees that your tests are more reliably executed when you use Arquillian.

The same applies to the TomEE ApplicationComposer, which is an extension of the OpenEJB embedded solution. It allows you to define more components and dependencies that need to be available at runtime during your tests. This flexibility has the downside that you carefully need to assemble and define all the required parts to make your tests execute successfully.

IDEs

TomEE server is supported with the Eclipse, IntelliJ, and NetBeans IDEs. Payara Server is also supported in these tools through the Payara Plugins we have developed.

When you start using Payara Server, you can still perform all the actions, starting and stopping the server, deploy an application, debug your code, etc....from within your favorite IDE. We also have support for Visual Studio Code in case you want to try out the newest IDE targeted for Java Applications.

TomEE vs. Payara Server Comparison

The TomEE server brings you the Java EE and Jakarta EE specifications on top of the Tomcat server. The server does not have many additional features on top of what is required by the Enterprise specifications. Migration to the Payara Platform is very straightforward as they support the same specifications. TomEE has some functionality around testing that does not have a direct counterpart within Payara Server. Some changes need to be introduced in your test when you migrate but the result is that you run on a platform that has far better monitoring, configuration capabilities, and is up to date with the latest standards. Payara Server has an extensive Monitoring Service that keeps track of the submodules and has a diverse set of notification and alerting channels. All this can be configured using the Asadmin CLI tool that makes the configuration automatable through scripts and it does not rely on manual configuration through XML files with a complex format.

Feature	TomEE 8	Payara Server Enterprise 5
Licence	Open Source	Open Source
Release frequency	Irregular	Monthly
Releases in 2019	3	22: 4 community stream, 12 stability stream, 8 feature stream
Patch releases	Irregular	Monthly (as of June 2020)
Security fixes	Infrequent	• Instant emergency and backported fixes for Payara Enterprise customers • As soon as possible for community
Production Support	✓	✓
Component Upgrades (e.g. Tyrus, Mojarra)	Irregular	Monthly
Supported IDEs	• Eclipse • Netbeans • IntelliJ IDEA • Visual Studio Code	• Eclipse • Netbeans • IntelliJ IDEA • Visual Studio Code
Caching tools	JCache	JCache, Domain Data Grid, Payara Scales (additional cost)
Automatic Clustering	No, manual via Tomcat Cluster	✓ (via Hazelcast)
Command line tool	✗	✓ Asadmin
Web Console	Basic (From Tomcat)	Almost all commands supported
Asadmin command recorder	✗	✓
Slow SQL Logging	✗	✓
Healthcheck Service	✗	✓
Request Tracing	✗	✓
Monitoring Logging	✗	✓
Microservices Distribution	Yes, TomEE MicroProfile	✓ (Payara Micro Only)

Feature	TomEE 8	Payara Server Enterprise 5
MicroProfile Support	MicroProfile 2.2	MicroProfile 3.2 (5.193) MicroProfile 3.3 (5.2020.2 / 5.20.0)
Docker Support	Example Images (no image for every release)	Official Images
Generate Uber JAR	✓	✓ (Payara Micro Only)
Production-tuned domain template	✗	✓
EE Standard	Jakarta EE 8	Jakarta EE 8
Jakarta EE Compatible	✗	✓
Runs on JDK 11	✓ (not mentioned in release doc but they have Docker images)	✓

For more migration resources to help you plan your migration or to learn how Payara can assist you with your migration, visit our online Migration Guide.

sales@payara.fish **+44 207 754 0481** **www.payara.fish**