

JAVAX TO JAKARTA TALES FROM THE CRYPT

Oct 10, 2023
Community / Code
Halifax, Nova Scotia

Session Objectives

- Learn about
 - Javax-to-Jakarta migration
 - Who, What, When, Where and Why

More Sessions Sun, Oct 8

- Tomcat 11 and Jakarta EE 11 Preview
 - Mark Thomas
 - 12:10 pm
- Why Apache Tomcat Matters for Jakarta EE
 - Ivar Grimstad
 - 2:20 pm
- From Spring Boot 2 to Spring Boot 3 with the Help of Apache Tomcat and More
 - Ivar Grimstad
 - 4:10 pm
 - Sample code for migrating to Spring 6 / Spring Boot 3:
<https://github.com/ivargrimstad/jakartaee-spring>



Shawn McKinney

@shawnmckinney

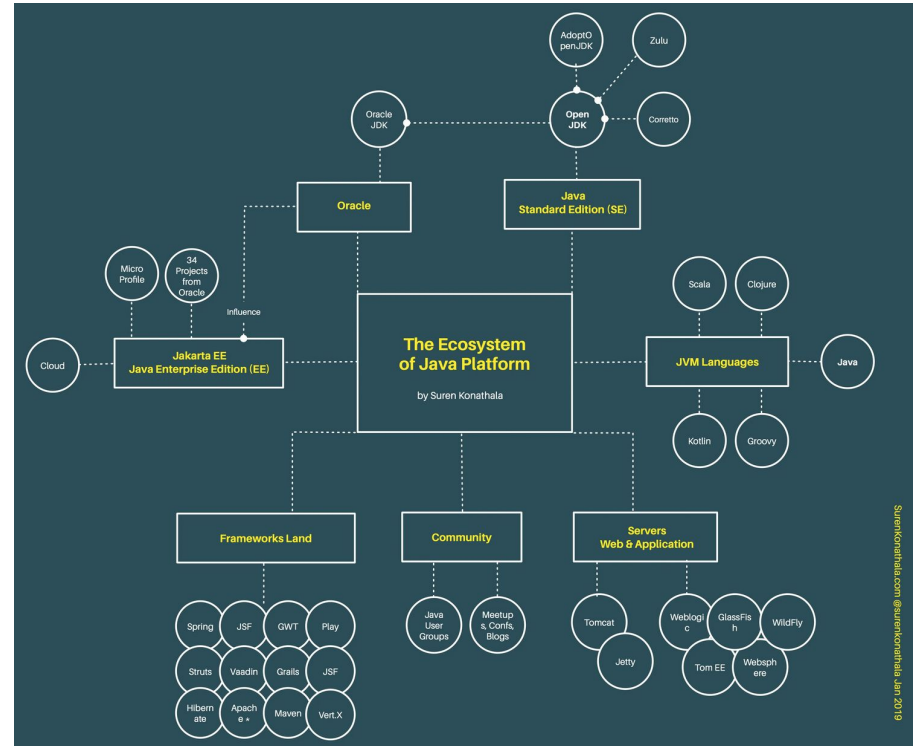
Being an aging cyclist is a bit like an old coder. Velocity and endurance decline, experience compensates. We choose better lines and make fewer mistakes.

- Works at <https://www.symas.com/>
- ASF member, Directory PMC <https://directory.apache.org/>
- Maintainer Apache Fortress <https://directory.apache.org/fortress/>
- Specializations
 - Application security, LDAP, Ansible
- Interests
 - Cycling & Open Source <https://iamfortress.net/>
 - Bike Monkey <https://twitter.com/ABikeMonkey>

Reuse

For many years now we've had this durable base of auxiliary software coming from the Java ecosystem. It's free and shields us from the intricacies and accelerates development. Makes our job easier.

- Upgrade from javax to jakarta — not yet!
<https://iamfortress.net/2022/08/13/upgrade-from-java-x-to-jakarta-not-yet/>



Ecosystem



Shawn McKinney

@shawnmckinney



The comparative value of a proposed computing language rests with its ecosystem. That's to say without a strong ecosystem, the language is not a good candidate for use. Why (what most of us would consider) a bad language does well in spite of its caveats. Not calling anyone out.

8:07 AM · Aug 15, 2022

Platform

The programming language itself factors into our choice. Java ranks at or near the top, in my estimation. We're not discussing theory or practice here and I'm not trying to convince you the platform's viable.

- Upgrade from javax to jakarta — not yet! <https://iamfortress.net/2022/08/13/upgrade-from-javax-to-jakarta-not-yet/>



Cost

There's a cost associated with using free software, despite the language in which it was written. It changes. Always for reasons outside of our control. That means we must change our code to accommodate. The Jakarta migration requires that millions (billions?) of lines-of-code be changed. We're talking about the biggest disruption to the Java ecosystem in its 28 years.

- Upgrade from javax to jakarta — not yet! <https://iamfortress.net/2022/08/13/upgrade-from-javax-to-jakarta-not-yet/>

What's the Problem



Shawn McKinney

@shawnmckinney



After many years we in the Java community have enjoyed remarkable stability. That being said, raise your hand if you didn't see the javax-to-jakarta clusterfuck coming. Don't get me wrong, I'm not laying blame (and what good would it do) but can you say line of demarcation?

8:47 AM · May 16, 2022

Disruption

The javax to jakarta namespace switch is one of the biggest changes into Java of the latest 20 years. No doubt. The entire ecosystem is impacted. Not only Java EE or Jakarta EE Application servers, but also libraries of any kind (Jackson, CXF, Hibernate, Spring to name a few).

- [From javax to jakarta](<https://tomee.apache.org/javax-to-jakarta.html>)

Specifications

- Web
 - Servlets
- Web service
 - JSON, REST, SOAP
- Enterprise
 - Beans, Messaging, Persistence, Transactions
- Runtimes
 - Apache Tomcat, Glassfish, JBoss

Java Enterprise Platform History

- JDK: '95 - '99
- J2EE 1.2 - 1.4* ● J2EE: '99 - '03
- JAVAE 5 - 8* ● JavaEE: '03 - '19
- JAKARTAEE 9, 10, ...* ● JakartaEE: '19 - present day

Namespace Change

In Java programs, dependencies are referenced via package names.

- Package names changed from `javax.*` to `jakarta.*`

```
import javaxJAKARTA.xml.bind.annotation.XmlAccessType;
```



Shawn McKinney


@shawnmckinney



Java's package naming convention is effective for references in code and pulling binary dependencies from the repos. The drawback's found when a name changes for a common package, say logging.

7:58 AM · Jan 19, 2023 · **121** Views

Name Change: Package X → Y

- Project A uses X + B
 - Project B uses X + C
 - Project C uses X
 - X changes name to Y
 - A uses Y + B *WAITING ON*
 - B uses Y + C *WAITING ON*
 - C uses X *HAS NOT MIGRATED*
- 

Why did the Javax namespace have to change?

Java EE was maintained by Oracle under the Java Community Process. On September 12, 2017, Oracle Corporation announced that it would submit Java EE to the Eclipse Foundation.

- [Jakarta EE](https://en.wikipedia.org/wiki/Jakarta_EE)

So what?



Shawn McKinney

@shawnmckinney

"I try keep up with the dependents but what worries me are the dependencies."

-- Unknown Github Dev'pr

What is The Eclipse Foundation?

An Independent, Europe-based not-for-profit corporation that acts as a steward of the Eclipse open source software development community, with legal jurisdiction in the European Union. It is an organization supported by over 350 members, and represents the world's largest sponsored collection of Open Source projects and developers. The Foundation focuses on key services such as intellectual property (IP) management, ecosystem development, and IT infrastructure.

- [Eclipse Foundation](https://en.wikipedia.org/wiki/Eclipse_Foundation)

What is Jakarta EE?

Formerly known as Java Platform, Enterprise Edition (Java EE) and Java 2 Platform, Enterprise Edition (J2EE), is a set of specifications, extending Java SE with specifications for enterprise features such as distributed computing and web services. Jakarta EE applications are run on reference runtimes, that can be microservices or application servers, which handle transactions, security, scalability, concurrency and management of the components they are deploying.

- [Jakarta EE](https://en.wikipedia.org/wiki/Jakarta_EE)

APACHE TOMCAT

e.g.

- ~~Java~~ **JAKARTA** Servlet
- ~~Java~~ **JAKARTA** Server Pages (JSP)
- Enterprise ~~Java~~ **JAKARTA** Beans (EJB)
- Contexts and Dependency Injection (CDI)
- ~~Java~~ **JAKARTA** API for Restful Services (JAX-RS)
- ~~Java~~ **JAKARTA** Architecture for XML Binding (JAXB)
- Websocket
- ...

The Crux

Oracle's Java trademarks are the property of Oracle and the Eclipse Foundation has no rights to use them. The implications of this are as follows:

- 1. The javax package namespace may be used within Jakarta EE specifications but may be used “as is” only. No modification to the javax package namespace is permitted within Jakarta EE component specifications. Jakarta EE specifications that continue to use the javax package namespace must remain TCK compatible with the corresponding Java EE specifications.*
- 2. Jakarta EE component specifications using the javax package namespace may be omitted entirely from future Jakarta EE Platform specifications.*
- 3. Specification names must be changed from a “Java EE” naming convention to a “Jakarta EE” naming convention. This includes acronyms such as EJB, JPA or JAX-RS.*

In addition to the above, any specifications which use the javax namespace will continue to carry the certification and container requirements which Java EE has had in the past. I.e., implementations which claim compliance with any version of the Jakarta EE specifications using the javax namespace must test on and distribute containers which embed certified Java SE implementations licensed by Oracle. These restrictions do not apply to Jakarta EE specifications which do not utilize javax, including future revisions of the platform specifications which eliminate javax.

- Update on Jakarta EE Rights to Java Trademarks
- May 3, 2019
- <https://blogs.eclipse.org/post/mike-milinkovich/update-jakarta-ee-rights-java-trademarks>

One option facing the Foundation is to simply give up on the negotiations, and declare that after 18+ months, the relationship with Oracle relating to Jakarta EE is dysfunctional and therefore it is not possible to reach agreement. He explained that, in the worst case, this would mean Jakarta EE would be terminated.

- [Eclipse Foundation Board meeting minutes](https://www.eclipse.org/org/foundation/boardminutes/2019_03_26_Minutes.pdf)

Why the Namespace Matters

- Non-migrated code doesn't work with newer container/runtime.
- Older container/runtime doesn't work with newer (migrated) code.
- There are workarounds. Conversion routines that modify the packaged bytecode so it works with the jakarta namespace. *SEARCH AND REPLACE*

How

- Change the source **BREAKS BACKWARD COMPATIBILITY**
 - Change maven coordinates (group ID changed from `javax.*` to `jakarta.*`)
 - Modify import statements in code
 - Modify properties prefixed with `javax`.
 - Upgrade to latest JVM/containers
- Change the binaries **NOT A GOOD OPTION FOR LIBS LIKE APACHE FORTRESS CORE**
 - Eclipse transformer: <https://projects.eclipse.org/projects/technology.transformer>
 - Tomcat migration tool: <https://tomcat.apache.org/download-migration.cgi>
 - IntelliJ Migrating to `jakarta` namespace:
<https://www.jetbrains.com/idea/guide/tutorials/migrating-javax-jakarta/>
 - Open Rewrite Jakarta recipes: <https://docs.openrewrite.org/recipes/java/migrate/jakarta>
- Both **OK FOR APPS**
 - Modify project source
 - Modify binaries of non-migrated dependencies

Change the Source *MAVEN COORDINATES*

```
<dependency>
  <groupId>javax.xml.bind</groupId>
  <artifactId>jaxb-api</artifactId>
  <version>${version.jaxb.api}</version>
</dependency>
```

To:

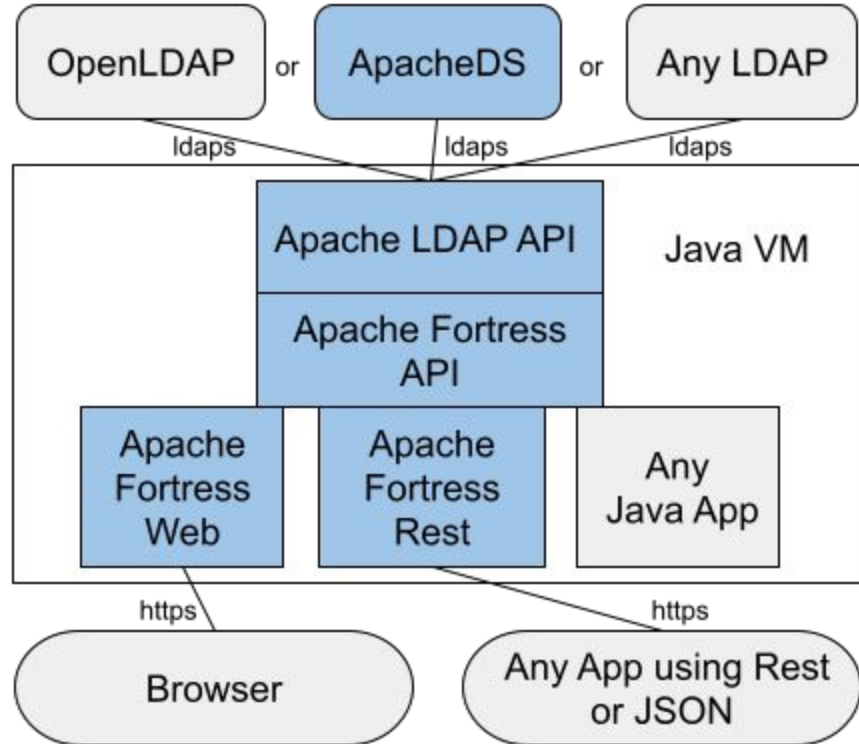
```
<dependency>
  <groupId>jakarta.xml.bind</groupId>
  <artifactId>jakarta.xml.bind-api</artifactId>
  <version>${version.jaxb.api}</version>
</dependency>
```


Change the Source *PACKAGE IMPORTS*

```
src/main/java/org/apache/directory/fortress/core/model/AuthZ.java @@ -20,10 +20,10 @@  
20 package org.apache.directory.fortress.core.model;  
21  
22  
23 - import javax.xml.bind.annotation.XmlAccessType;  
24 - import javax.xml.bind.annotation.XmlAccessorType;  
25 - import javax.xml.bind.annotation.XmlRootElement;  
26 - import javax.xml.bind.annotation.XmlType;  
23 + import jakarta.xml.bind.annotation.XmlAccessType;  
24 + import jakarta.xml.bind.annotation.XmlAccessorType;  
25 + import jakarta.xml.bind.annotation.XmlRootElement;  
26 + import jakarta.xml.bind.annotation.XmlType;
```

Apache Fortress

- Subproject of Apache Directory
- Written in Java
- Role-Based Access Control
- Four Components
 - Core: Java APIS **JAXB**
 - Realm: Java EE security
 - Web: Administrative UI **SERVLET**
 - Rest: APIs/HTTP **JAX-RS**
- [Apache Fortress](<https://directory.apache.org/fortress/>)



Three Tries

Stale branches

`trivial/support-ha-client` Updated 5 years ago by yudhik

`FC-247` Updated 5 years ago by kayyagari

`jakarta` Updated last year by shawnmckinney

`jakarta2` Updated last year by shawnmckinney

`jakarta3` Updated 4 months ago by shawnmckinney

Consequences **ALL OR NOTHING**

- *Every API must transition the Java package name from javax to jakarta.*
- *New versions of app servers must be released. The new impls won't work on Java EE-based app servers and the old impls won't work on Jakarta EE-based app servers.*
- *Jakarta EE 8 APIs do not work with Jakarta EE 9 impls and vice versa.*

- [Understanding Jakarta EE 9](https://www.eclipse.org/community/eclipse_newsletter/2020/november/1.php)

Strike One



Shawn McKinney

@shawnmckinney



Spent a w/e being battered by an ill-fated javax-to-jakarta conversion of jaxb libs. There were false starts / upgrades. There were hard resets and reverts. I even had to consult stack overflow posts. In the end, a full retreat. You may have won this time jaxb, but I'll be back.

10:11 AM · Mar 20, 2022

Reflection



Shawn McKinney

@shawnmckinney

How about this one? (asking for a friend)

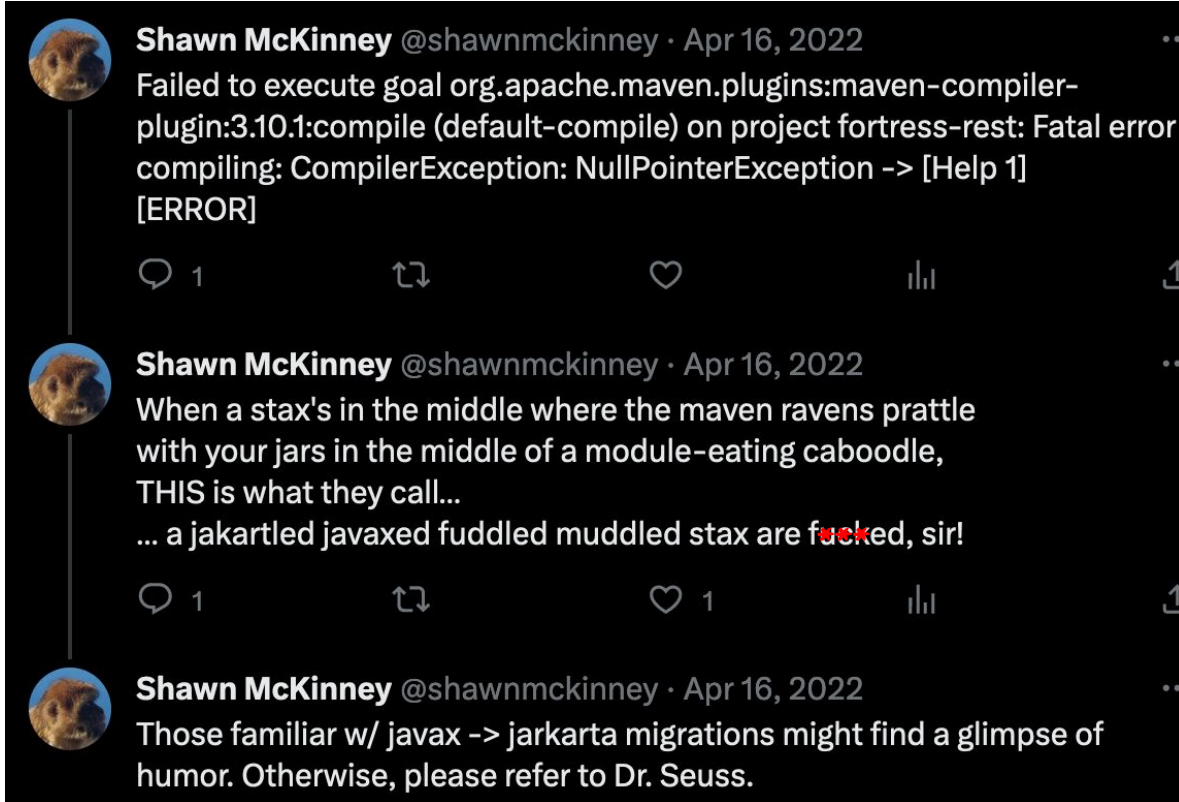
Jax be impl,

Jax be quick,


Jax be dumping out a fricken stacktrace during my Javax-to-Jakarta migration

12:53 PM · Apr 17, 2022

Acceptance




Strike Two

 **Shawn McKinney**
@shawnmckinney

Written after a second scuttled[1] attempt to migrate a JAXB library to Jakarta. (Not a rant post)
iamfortress.net/2022/08/13/upg...

[1]transitive verb
: to cut a hole through the bottom, deck, or side of (a ship)
specifically : to sink or attempt to sink by making holes through the bottom

 iamfortress.net
Upgrade from javax to jakarta — not yet!
It's now been almost five years since Oracle announced the donation of Java's enterprise layer to the Eclipse ...

5:58 AM · Jan 13, 2023 · **100** Views

Update

April 24, 2023

Many of the suppliers are getting their migrated packages released. For example, Spring 6, Apache CXF 4 and Apache Wicket 10.



Shawn McKinney

@shawnmckinney

Just got a successful test using Jakarta (instead of Javax) w/ Apache Tomcat 10, Apache Fortress 2.0.9, Apache CXF 4 and Spring 6! Started this epic journey about 2 years ago.

8:12 PM · Apr 23, 2023 · **186** Views

Third time's the charm



Shawn McKinney

@shawnmckinney



The logjam has busted up. Apache Fortress now migrates to Jakarta. So long Javax. You served us well and your time has passed.



Apache Fortress @apache_fortress · Jun 19

Can you say Jakarta compatible? Just in time for our 3.0 release!
[twitter.com/apache_wicket/...](https://twitter.com/apache_wicket/)

9:34 AM · Jun 19, 2023 · **127** Views

Logjam

The runtime's ready, with a logjam in the projects upstream. Can't migrate until all dependencies have published compatible packages. In turn dependent projects must wait until we publish our conversion, and so on. We can replace non-compliant libraries (a ton of work) or wait.

- Upgrade from javax to jakarta — not yet! <https://iamfortress.net/2022/08/13/upgrade-from-javax-to-jakarta-not-yet/>



More Info

- Migrating from javax to jakarta namespace
https://youtu.be/mukr2Q_zBm4?si=JQmJvszQHH-V5zpH
- Transition from Java EE to Jakarta EE
<https://blogs.oracle.com/javamagazine/post/transition-from-java-ee-to-jakarta-ee>
- From javax to jakarta namespace <https://tomee.apache.org/javax-to-jakarta.html>

NOT A RANT

That has all been covered elsewhere. I will point out that converting the code is simple. Just change the package imports and pull in the newer versions of the libs into whatever dependency mechanism is being used.

If you can find all of them. As of today, many affected suppliers have published Jakarta compatible releases, some are still working on it, while still others have it on their roadmap.

That's quite remarkable actually and demonstrates the robustness of the ecosystem. It'll handle it — eventually.

- [Upgrade from javax to jakarta — not yet!](<https://iamfortress.net/2022/08/13/upgrade-from-javax-to-jakarta-not-yet/>)

Parting thoughts

- We (the ecosystem) aren't going away
- Jakarta's in good hands
- Here's to 20 more years of stability

Contact

Shawn McKinney

- Email: smckinney@apache.org
- Twitter: @shawnmckinney
- Blog: <https://www.symas.com/symas-blog>
- Project: <https://directory.apache.org/fortress/>

