

# JSR-222 Maintenance Review (JAXB) 2.3

## **Description:**

Maintenance review for the JAXB 2.3 specification

## **Maintenance Lead:**

Roman Grigoriadi, Oracle Corporation

## **Feedback:**

Comments can be send to [users@jaxb.java.net](mailto:users@jaxb.java.net)

## **Rationale of changes:**

Make service discovery of JAXB implementation compatible with JDK9 module system.

Proposed changes:

### **1. Make JAXB API runnable as JDK9 module**

JAXB API now can be loaded as JDK 9 module. A JDK 9 module descriptor `module-info.java` has been added. If run on JKD 9 and loaded from module path, JAXB API will be defined as a module. JAXB API module name is set to `"java.xml.bind"`, which is the same as module name in JDK containing api and implementation. If used as a module, JDK 9 bundled `"java.xml.bind"` must be upgraded (`--upgrade-module-path`) with standalone JAXB API module. This approach will replace the former endorsed mechanism in module system environment.

### **2. Update discovery of implementation**

Process of discovering implementation has changed, in order to be compatible with JDK 9 module system. All old methods of discovery are backwards compatible and supported, the following discovery method has been added:

New interface - **`javax.xml.bind.JAXBContextFactory`** has been added to support `java.util.ServiceLoader` mechanism. This is now the preferred approach, which implementations should provide to be discovered (described in step 3).

"Manual service provider" described in step 4 is now deprecated in favor of step 3 approach.

Implementation discovery order has changed to the following (first successful resolution applies):

1. Look for `jaxb.properties` in a package of `"classesToBeBound"` / `"contextPath"` parameters passed to `JAXBContext#newInstance`. If found, `"jaxb.properties"` are loaded as a resource and value for `"javax.xml.bind.JAXBContextFactory"` key or `"javax.xml.bind.context.factory"` key is used as provider factory class.
2. System properties `"javax.xml.bind.JAXBContextFactory"`, `"javax.xml.bind.context.factory"`, and `"javax.xml.bind.JAXBContext"` are checked in the order specified. If value is set it's used as provider factory class.
3. Load `javax.xml.bind.JAXBContextFactory` implementation with `java.util.ServiceLoader`. The service-provider loading facility will use the current thread's context class loader to attempt to load the context factory implementation. If the context class loader is null, the system class loader will be used.
4. Look for resource `/META-INF/services/javax.xml.bind.JAXBContext`. If exists, its content is assumed to be the provider factory class (this approach is now deprecated).
5. If all the steps above fail, then the rest of the lookup is unspecified.

Considering new interface `"javax.xml.bind.JAXBContextFactory"` and its support for service provider mechanism, following has been removed:

SPEC REQUIREMENT: the provider must supply an implementation class containing the following method signatures:

```
public static JAXBContext createContext( String contextPath, ClassLoader  
classLoader, Map<String, Object> properties ) throws JAXBException
```

```
public static JAXBContext createContext( Class[] classes, Map<String, Object>  
properties ) throws JAXBException
```

### **3. New restrictions, applicable only when JDK9 module system is available:**

JAXB annotated classes (classesToBeBound and contextPath parameters of JAXBContext#newInstance methods), which are accessed by JAXB runtime, are now required to be open (java.lang.reflect.Module#isOpen String, Module) to "java.xml.bind" module in order to be accessible by reflection.

If JAXB implementation is defined in other module than "java.xml.bind", than API will propagate openness of annotated classes to implementation module.

JAXB API is now packaged as JEP 238: Multi-Release JAR, in order to access JDK9 Module related API without reflection.

### **4. Minor javadoc changes**

JAXBContext class level javadoc, "Unmarshalling" section:

"should" changed to "must" in sentence describing merging of global elements across set of schemas.

---

Method level javadoc for all variants of "JAXBContext.newInstance" methods, have changed "Obtain" to "Create".

---

javadoc of throws clause of newInstance(java.lang.Class...), newInstance(java.lang.String) and newInstance(java.lang.String, java.lang.ClassLoader) has been replaced with a link to newInstance(java.lang.Class[], java.util.Map) / newInstance(java.lang.String, java.lang.ClassLoader, java.util.Map) throws clause.

---

java.xml.bind.JAXBElement and java.xml.bind.helpers.AbstractMarshallerImpl

In method level javadoc of JAXBElement#marshal(\*) methods, throws section - fixed literal representing argument of an object to marshall from "obj" to "jaxbElement".

---

java.xml.bind.ValidationEvent

Fixed javadoc of ValidationEvent#getSeverity() - "ValidationError" replaced with "ValidationEvent".

---

Fixed minor typo javadoc errors in `java.xml.bind.annotation` in classes `XmlElements`, `XmlIDREF`, `XmlSchema`,

---

"`package.html`" files was interchanged with "`package-info.java`". Links in Javadoc have been fixed.

---

Detailed javadoc diff changelist can be found in `javadoc-diff.tar.gz` file.