



Vánoční hrátky s JAXB

<http://jaxb.dev.java.net>

Martin Grebáč, martin.grebac@sun.com

Pavel Bucek, pavel.bucek@sun.com



Přehled

- Příklad 1 - Jednoduché čtení/zapisování z/do XML
- Co je JAXB
 - > O projektu
 - > Framework
- Příklad 2 - XML schéma, generování Java kódu, Validace
- Možnosti rozšíření
- Použití
- Kde, co a jak
- Otázky

DEMO1 – Rychle, jak můžu psát a číst tohle XML?

- Jak co nejrychleji začít s JAXB

```
public class Gift {  
    public enum Color { ... }  
    public String recipient;  
    public int size;  
    public Color color;  
  
    public Gift(recipient, size, color) { ... }  
}
```

JAXB – Historie

- JAXB 1
 - > mapování dat pro framework webových služeb JAX-RPC
 - > součástí Java EE 1.4
- JAXB 2.0 - “Velký třesk”
 - > Samostatný standard – JSR 222 (<http://www.jcp.org>)
 - > Velké změny architektury a vylepšení (anotace, mnohem méně generovaného kódu, vyšší výkon)
 - > Mapování dat pro api pro webové služby JAX-WS 2.0
 - > Integrace do Javy – přímá součást JDK 6
- JAXB 2.1
 - > JDK 1.6 update 4, JAX-WS 2.1

JAXB – Současnost

- JAXB 2.2
 - > 10.12.2009
 - > Plánovaná integrace do JDK 7
 - > Součástí Java EE 6
 - GlassFish V3 – referenční implementace
 - > JAX-WS 2.2 integrace

JAXB – O Projektu

- Spec lead: Kohsuke Kawaguchi
- Tech lead: Martin Grebáč
- Developer: Pavel Bucek
- TCK, JCK: Leonid Kuskov
- Licencee engineer: David Korbel
- Everything else (QE/RE/Docs/Webadmin): Martin/Pavel

Ratings & Reviews

Community Rating


 4.3/5.0

Ohloh Analysis Summary

-  [Mostly written in Java](#)
-  [Mature, well-established codebase](#)
-  [Increasing year-over-year development activity](#)
-  [Well-commented source code](#)

Updated 01 Dec 2009 19:17 UTC

[Homepage](#)

<https://jaxb.dev.java.net/>

[Licensed under](#)

[Common Development and Distribution License](#)

[Tagged as](#)

 [Add tags to this project](#)

DEMO 2 – co s tímhle .xsd souborem?

- Jak používat JAXB s XSD schématem

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://xml.netbeans.org/schema/czjug02"
  xmlns:tns="http://xml.netbeans.org/schema/czjug02"
  elementFormDefault="qualified">

  <xsd:element name="christmasTree"><xsd:complexType><xsd:sequence>
    <xsd:element name="gift" minOccurs="0" maxOccurs="20">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="recipient" type="xsd:string" />
          <xsd:element name="size" type="xsd:int" />
          <xsd:element name="color" type="tns:color" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence></xsd:complexType></xsd:element>

  <xsd:simpleType name="color"><xsd:restriction base = "xsd:string">
    <xsd:enumeration value = "RED" />
    ...
  </xsd:restriction></xsd:simpleType>

</xsd:schema>

```

DEMO 2 – ještě jednou, jak ?

> xjc christmas.xsd (vygeneruje ChristmasTree.java, ...)

```
JAXBContext ctx = JAXBContext.newInstance("package");
```

Zápis do XML

```
ChristmasTree myTree = new ChristmasTree();
```

```
myTree.set...;
```

```
Marshaller m = ctx.createMarshaller();
```

```
m.marshal(myTree, f);
```

Čtení z XML

```
Unmarshaller u = ctx.createUnmarshaller();
```

```
ChristmasTree myTree = (ChristmasTree) u.unmarshal(is);
```

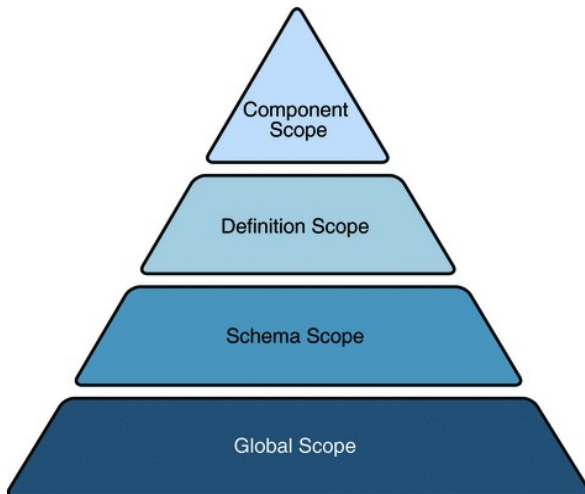

JAXB – Úpravy, nastavení a rozšíření

- Vnořené / Externí reference

```
<xs:annotation>
  <xs:appinfo>
    ... binding declarations ...
  </xs:appinfo>
</xs:annotation>
```

```
> xjc -b bindingfile schema
```

- Globální / Lokální



```
<jaxb:property ... />
```

```
<jxb:class ... />
```

```
<jxb:schemaBindings ... />
```

```
<jxb:globalBindings ... />
```

Globální úpravy

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:jxb="http://java.sun.com/xml/ns/jaxb"
            jxb:version="2.0">
  <xsd:annotation>
    <xsd:appinfo>
      <jxb:globalBindings
        collectionType="java.util.Vector"
        choiceContentProperty = boolean
        enableFailFastCheck = boolean
        enableJavaNamingConventions = boolean
        enableValidation = boolean
        fixedAttributeAsConstantProperty = boolean
        generateElementClass = boolean
        generateElementProperty = boolean
        generateValueClass = boolean
        localScoping = "nested" | "toplevel"
        mapSimpleTypeDef = boolean
        optionalProperty = "wrapper" | "primitive" | "isSet"
        underscoreBinding = "asWordSeparator" | "asCharInWord"
      />
    </xsd:appinfo>
  </xsd:annotation>
```

Globální úpravy – xjc:simple (proprietary)

- Přesnější generování typů
- @XmlRootElement
- Generování (anglických) plurálů
- Cíl: ještě větší zjednodušení práce s JAXB
- Experimentální, proprietární

Globální úpravy – xjc:simple (experimental)

```
<xs:schema xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
  jaxb:extensionBindingPrefixes="xjc">
  <xs:annotation> <xs:appinfo>
    <jaxb:globalBindings generateValueClass="false">
      <xjc:simple />
    </jaxb:globalBindings>
  </xs:appinfo> </xs:annotation>
  <xs:complexType name="Person">
    <xs:choice>
      <xs:sequence>
        <xs:element name="a" type="xs:int" />
        <xs:element name="b" type="xs:int" />
      </xs:sequence>
      <xs:sequence>
        <xs:element name="b" type="xs:int" />
        <xs:element name="c" type="xs:int" />
      </xs:sequence>
    </xs:choice>
    <xs:sequence>
      <xs:element name="child" type="xs:string" maxOccurs="unbounded"/>
      <xs:element name="parent" type="xs:string" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType></xs:schema>
```



```
// bez xjc:simple
public class Person {
    protected List<String> child;
    protected List<String> parent;
    List<JAXBElement<Integer>> content;
}

// s <xjc:simple> nastavením
@XmlRootElement
public class Person {
    protected List<String> children;
    protected List<String> parents;
    Integer a;
    int b;
    Integer c;
}
```

Lokální úpravy

```

<xsd:complexType name="ChristmasGiftsType">
  <xsd:annotation>
    <xsd:appinfo>
      <jxb:class name="GiftType">
        <jxb:javadoc>
          A <b>Gift</b> consists of addresses and items.
        </jxb:javadoc>
      </jxb:class>
    </xsd:appinfo>
  </xsd:annotation>
  ...

```



```

/**
 * A <b>Gift</b> consists of addresses
 * and items.
 */
@XmlType(name="ChristmasGiftsType")
public class GiftType {
    ...
}

```

Rozšíření - XJC pluginy

- <https://jaxb2-commons.dev.java.net/>
 - Návod k použití:
 - > Stáhnout jar pluginu
 - > Spustit xjc s jarem pluginu na classpath a identifikátorem (názvem) pluginu
- > `xjc -classpath plugin.jar -Xplugin ...`

Rozšíření - XJC pluginy – Basic Set

- <https://jaxb2-commons.dev.java.net/>



JAXB2 Basics Plugins

[View](#) · [Attachments \(0\)](#) · [Info](#)

Introduction

Schema compiler (XJC) produces schema-derived classes which can be used to turn XML into object structures and back. However, generated classes lack by default convenience and utility methods like `equals(...)`, `hashCode(...)`, `toString()` and so on.

JAXB2 Basics provides a package of plugins which can generate such utility code

Plugins

- [Equals plugin](#)
- [HashCode plugin](#)
- [ToString plugin](#)
- [Copyable plugin](#)
- [Mergeable plugin](#)
- [JAXBIndex plugin](#)
- [Inheritance plugin](#)

Rozšíření - XJC pluginy – Fluent API

- <https://jaxb2-commons.dev.java.net/>

```
USAddress address = new USAddress();  
address.setName( name );  
address.setStreet( street );  
address.setCity( city );  
address.setState( state );  
address.setZip( new BigDecimal( zip ) );
```

```
USAddress address = new USAddress()  
    .withName(name)  
    .withStreet(street)  
    .withCity(city)  
    .withState(state)  
    .withZip(new BigDecimal(zip));
```


Rozšíření - XJC pluginy – Annotations

- <https://jaxb2-commons.dev.java.net/>

```
<xsd:annotation>
  <xsd:appinfo>
    <annox:annotate>
      <jl:SuppressWarnings/>
    </annox:annotate>
    <annox:annotate target="setter">
      <jl:Deprecated/>
    </annox:annotate>
    <annox:annotate target="getter">
      <jl:Deprecated/>
    </annox:annotate>
    <annox:annotate target="field">
      <jl:Deprecated/>
    </annox:annotate>
    <annox:annotate target="class">
      <jl:Deprecated/>
    </annox:annotate>
  </xsd:appinfo>
</xsd:annotation>
```

```
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "FooType", propOrder = {
    "bar",
    "foobar"
})
@SuppressWarnings({
})
@Deprecated
public class FooType {

    @XmlElement(required = true)
    protected String bar;
    @XmlElement(required = true)
    @Deprecated
    protected String foobar;
```

JAXB – Pro / proti JAXB

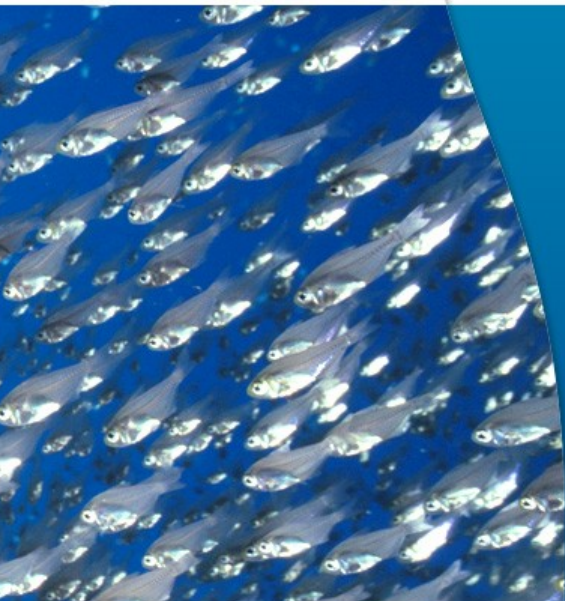
- Pro
 - > Zjednodušení vývoje aplikací s XML, zjednodušení a zdrojového kódu
 - > Pokud požadujete výkon a efektivní správu paměti
 - > Podpora streamování
 - > 100% podpora XSD formátu
 - > Váš framework již JAXB používá (Jersey, JAX-WS, Metro)
 - > Přímo součást JDK – podpora, standard, testy
 - > Spolehlivost a kvalita
 - Běžíme pravidelně několik desítek tisíc testů
 - Další velké sady testů běží na všech platformách podporovaných Javou při integracích do JDK / JAX-WS / Metra / Jersey

JAXB – Pro / proti JAXB

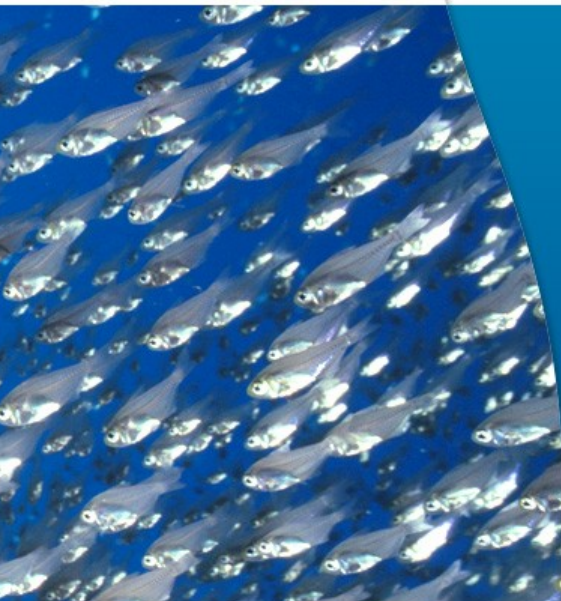
- Kdy zvážit alternativy
 - > JDK < 1.5
 - > Nepříliš striktně definovaná schémata
 - xs:any, xs:schema, “mixed” elementy, abstraktní typy a xsi definice, “nillable” elementy
 - > Kontrola nad přesným zápisem XML
 - dokumentace, odsazování, zachování formátování, ...
 - > Když standardní Java / XML mapování nevyhovuje
 - > Pokud kladete kritický důraz na čas strávený inicializací
 - Vytvoření instance JAXBContext je operace drahá na čas i paměť – JAXB hledá beany a vnáší do nich dynamicky metody nutné k přístupu k elementům

JAXB – Co jsme nestihli ukázat ...

- ... ale najdete to v dokumentaci nebo vzorových příkladech
- Java → Schéma přístup
- Částečný unmarshalling (Binder)
- Dynamické mapování XML elementů
- Jak řešit cyklické reference
- Typová substituce v XML/XSD, abstraktní typy
- Adaptéry (vlastní metody na zápis/čtení typů)
- Jak na kolize, katalogy, escaping znaků, DTD, JSON, mnoho dalších věcí ...



Otázky ?



<https://jaxb.dev.java.net>

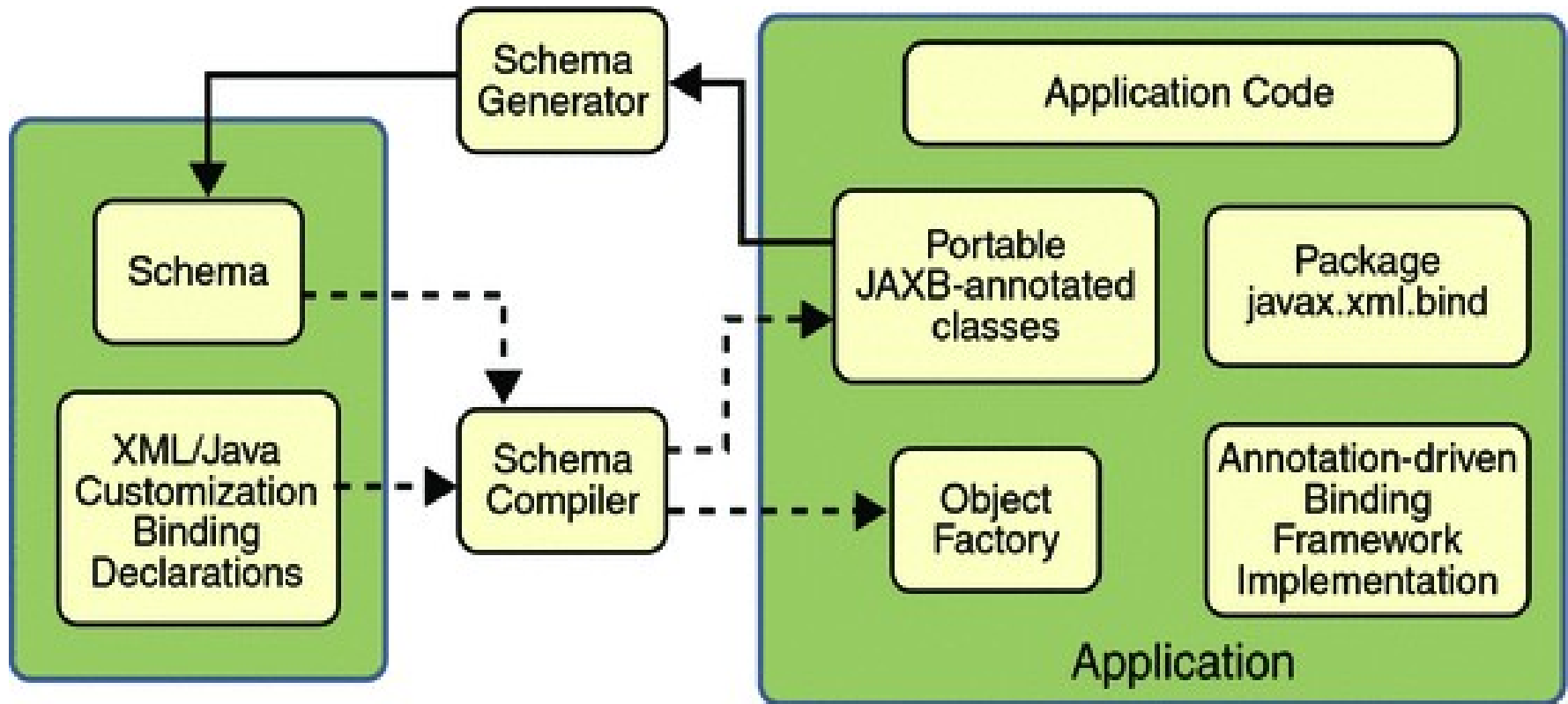
[https://jaxb2-commons.dev.java.net/
users@jaxb.dev.java.net](https://jaxb2-commons.dev.java.net/users@jaxb.dev.java.net)

<http://www.jcp.org/en/jsr/summary?id=222>

JAXB – Pojmy

- JAXBContext
- Marshaller/Marshalling
- Unmarshaller/Unmarshalling
- XJC
- Binding soubor
- XSD
- RelaxNG
- Validace

JAXB – Architektura



- - - - -> Schema to Java

————> Java to Schema