

Introduction to XDoclet

Erik Hatcher
eHatcher Solutions, Inc.

1

Goals

- Introduce XDoclet
- Understand its architecture
- Apply XDoclet to a J2EE project

2

What is XDoclet?

- Javadoc metadata templating engine
- Attribute-oriented programming
- Outgrown its EJBDoclet roots

3

Why XDoclet?

- Avoid code/metadata duplication
- Pragmatic Programming:
 - DRY - Don't Repeat Yourself
 - Program close to the problem domain
 - Write code that writes code
- JSR's 175 & 181

4

JSR 175

A metadata facility for the Java Programming Language would allow classes, interfaces, fields, and methods to be marked as having particular attributes.

5

JSR 181

This JSR defines an annotated Java format that that uses Java Language Metadata (JSR 175) to enable easy definition of Java Web Services in a J2EE container.

6

XDoclet Architecture

- Built upon XJavaDoc
- Separated into modules
- Embedded templates
- Sophisticated Ant tasks with dynamically loaded subtasks
- Tag handlers

Version 1.2

7

XJavaDoc

- JavaCC-based source code parser
- Builds model of:
 - Packages
 - Classes / inheritance hierarchy
 - Methods and parameters
 - Member variables
 - Constructors
 - Custom @tags
 - ... oh yeah, and Javadoc comments too

8

ToDo List

Overview Packages org.example.antbook.ant.lucene (4) org.example.antbook.common (1) org.example.antbook.web (1) Classes IndexTask (1) SearchInitServlet (1) SearchQueryException (1) FileExtensionDocumentHandler (1) TextDocument (2)	ToDo list for Generated by XDoclet. <table><tr><th>Location</th><th>Description</th></tr><tr><td>org.example.antbook.ant.lucene.FileExtensionDocumentHandler class</td><td>Implement dynamic document type lookup</td></tr><tr><td>org.example.antbook.ant.lucene.IndexTask m private void indexDocs()</td><td>refactor!!!!</td></tr><tr><td>org.example.antbook.ant.lucene.TextDocument class</td><td>Fix JavaDoc comments here</td></tr><tr><td>m public java.lang.String getContents()</td><td>finish this method</td></tr><tr><td>org.example.antbook.common.SearchQueryException class</td><td>Add printStackTrace and other constructors</td></tr><tr><td>org.example.antbook.web.SearchInitServlet class</td><td>Refactor to use JNDI for directory lookup.</td></tr></table>	Location	Description	org.example.antbook.ant.lucene.FileExtensionDocumentHandler class	Implement dynamic document type lookup	org.example.antbook.ant.lucene.IndexTask m private void indexDocs()	refactor!!!!	org.example.antbook.ant.lucene.TextDocument class	Fix JavaDoc comments here	m public java.lang.String getContents()	finish this method	org.example.antbook.common.SearchQueryException class	Add printStackTrace and other constructors	org.example.antbook.web.SearchInitServlet class	Refactor to use JNDI for directory lookup.
Location	Description														
org.example.antbook.ant.lucene.FileExtensionDocumentHandler class	Implement dynamic document type lookup														
org.example.antbook.ant.lucene.IndexTask m private void indexDocs()	refactor!!!!														
org.example.antbook.ant.lucene.TextDocument class	Fix JavaDoc comments here														
m public java.lang.String getContents()	finish this method														
org.example.antbook.common.SearchQueryException class	Add printStackTrace and other constructors														
org.example.antbook.web.SearchInitServlet class	Refactor to use JNDI for directory lookup.														

9

@todo

```
/**
 * Index the fileset.
 *
 * @exception IOException if Lucene I/O fails
 * @todo refactor!!!!
 */
private void indexDocs() throws IOException {
    // . . .
}
```

10

Ant <taskdef>

```
<taskdef name="todo"
         classname="xdoclet.modules.doc.DocumentDocletTask"
         classpathref="xdoclet.classpath"
/>

<target name="todo">
  <mkdir dir="${build.dir}/todo"/>

  <todo destdir="${build.dir}/todo" >
    <fileset dir="src/anttask"/>
    <fileset dir="src/common"/>
    <fileset dir="src/ejb"/>
    <fileset dir="src/web"/>

    <info/>
  </todo>
</target>
```

11

EJB Descriptor

```
.<!-- ejb-jar.xml -->
.
<enterprise-beans>

  <session >

    <ejb-name>org.example.antbook.session.SearchSession</ejb-name>

    <home>org.example.antbook.session.SearchSessionHome</home>
    <remote>org.example.antbook.session.SearchSession</remote>
    <ejb-class>
      org.example.antbook.session.SearchSessionBean
    </ejb-class>
    <session-type>Stateless</session-type>
    <transaction-type>Container</transaction-type>

  </session>

.
.
.
```

12

JBoss Descriptor

```
<jboss>

  <enterprise-beans>

    <session>
      <ejb-name>
        org.example.antbook.session.SearchSession
      </ejb-name>
      <jndi-name>
        personal/org.example.antbook.session.SearchSession
      </jndi-name>
    </session>

  </enterprise-beans>

  <resource-managers>
  </resource-managers>

</jboss>
```

13

Session Bean

```
/**
 * @ejb.bean type="Stateless"
 *   jndi-name="${site}/org.example.antbook.session.SearchSession"
 * @ejb.util generate="physical"
 */
public class SearchSessionBean implements SessionBean {
  //...

  /**
   * @ejb.interface-method
   */
  public Document[] search(String indexDir, String query)
    throws SearchQueryException, SystemException {
    return SearchUtil.findDocuments(indexDir, query);
  }
}
```

14

Ant Property Substitution

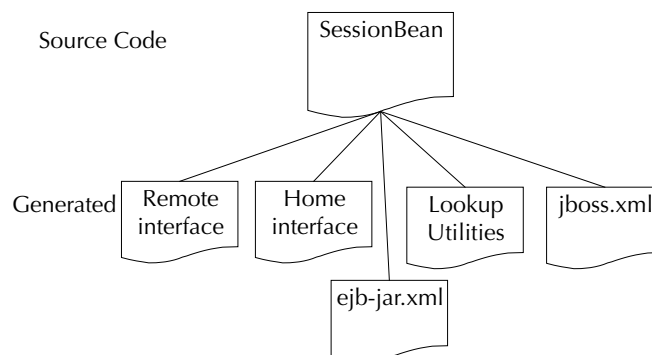
- Allows build-time control over values generated

```
<property name="site" value="sample"/>
```

```
/**  
 * @ejb.bean type="Stateless"  
 *   jndi-name="${site}/org.example.antbook.session.SearchSession"  
 * @ejb.util generate="physical"  
 */  
public class SearchSessionBean implements SessionBean { . . . }
```

15

Session Bean: XDoclet-style



16

<ejbdoclet>

```
<ejbdoclet destdir="${build.dir}/ejb/gen"
  addedtags="@xdoclet-generated at ${TODAY}"
  ejbspec="1.1"
  force="${xdoclet.force}"
  mergedir="metadata/ejb">

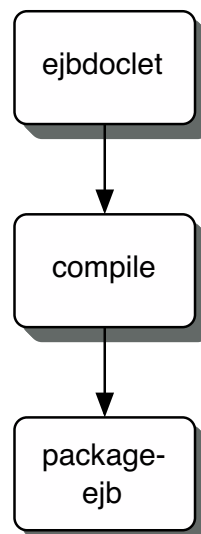
  <fileset dir="src/ejb"/>

  <remoteinterface/>
  <homeinterface/>
  <utilobject/>
  <jboss validatexml="true"
    destdir="${build.dir}/${site}"/>
  <deploymentdescriptor validatexml="true"/>

</ejbdoclet>
```

17

EJB build process



18

EJB build process

```
<target name="ejbdoclet" depends="init">
    <ejbdoclet...>
</target>

<target name="compile-ejb"
    depends="compile-common,ejbdoclet">
    <javac...>
</target>

<target name="package-ejb"
    depends="compile-ejb"
    <jar...>
</target>
```

19

Servlet Definition

```
<servlet>

    <servlet-name>search-init</servlet-name>

    <servlet-class>
        org.example.antbook.web.SearchInitServlet
    </servlet-class>

    <init-param>
        <param-name>index-dir</param-name>
        <param-value>/Users/erik/. . ./personal/index</param-value>
    </init-param>

    <load-on-startup>1</load-on-startup>

</servlet>
```

20

Servlet

```
/**
 * @web.servlet name="search-init"
 *             load-on-startup="1"
 * @web.servlet-init-param name="index-dir"
 *                       value="${index.dir}"
 *
 * @todo Refactor to use JNDI for directory lookup.
 */
public class SearchInitServlet extends HttpServlet {
    public void init() throws ServletException {
        super.init();

        ServletConfig config = getServletConfig();
        getServletContext().setAttribute(Constants.SEARCH_DIRECTORY,
            config.getInitParameter("index-dir"));
    }
}
```

21

Filter Definition

```
<filter>

    <filter-name>TimingFilter</filter-name>
    <filter-class>
        org.example.antbook.web.TimingFilter
    </filter-class>

</filter>

<filter-mapping>

    <filter-name>TimingFilter</filter-name>
    <url-pattern>/*</url-pattern>

</filter-mapping>
```

22

Filter

```
/**
 * Logs duration of each request.
 *
 * @web.filter name="TimingFilter"
 * @web.filter-mapping url-pattern="/*"
 *                  servlet-name="TimingFilter"
 */
public class TimingFilter implements Filter {
    // . . .
}
```

23

Tag Library Descriptor

```
<taglib>

  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>antbook</short-name>
  <tag>
    <name>buildprops</name>
    <tag-class>
      org.example.antbook.web.BuildPropertiesTag
    </tag-class>
    <tei-class>
      org.example.antbook.web.BuildPropertiesTei
    </tei-class>
  </tag>
</taglib>
```

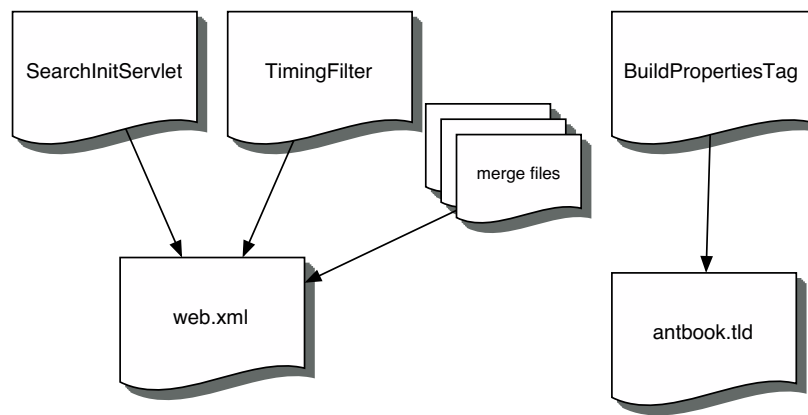
24

Tag Library

```
/**
 * Iterates build properties and places name/value into
 * page scope.
 *
 * @jsp.tag name="buildprops"
 *   bodycontent="JSP"
 *   tei-class="org.example.antbook.web.BuildPropertiesTei"
 */
public class BuildPropertiesTag extends BodyTagSupport {
    // . . .
}
```

25

<webdoclet> artifacts



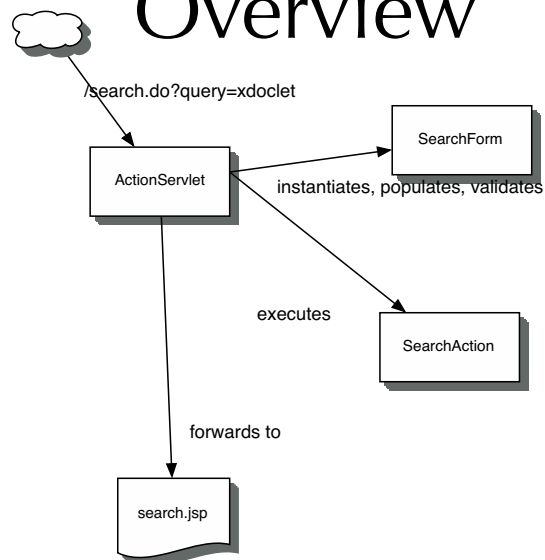
26

Merge Points

- Not everything is stored in source code
- Customization entry point to generated artifacts
- Often documented in the generated code
- web.xml, for example:
 - External servlet mappings
 - Tag library mapping
 - Welcome file list, error pages, etc

27

Quick Struts Overview



28

Struts Descriptors

- struts-config.xml
 - Global form bean definitions
 - Action mappings
- validation.xml
 - Field-level validations per form or action
 - I18N / L16N
 - Extensible

29

struts-config.xml

```
<form-beans>
  <form-bean name="SearchForm"
    type="org.example.antbook.struts.SearchForm"
  />
```

```
<!--
  If you have non XDoclet forms, define them in a file
  called struts-forms.xml and
  place it in your merge directory.
-->
```

```
</form-beans>
```

30

validation.xml

```
<form name="SearchForm">
  <field property="query"
    depends="required">

    <msg name="required"
      key="query.required"
    />

    <arg0 key="SearchForm.query"/>
  </field>
</form>
```

31

Struts ActionForm

```
/**
 * Search query entry form.
 *
 * @struts.form    name="SearchForm"
 */
public class SearchForm extends ValidatorForm {
    private String query;

    /**
     * Sets the query attribute of the SearchForm object
     *
     * @struts.validator type="required"
     *                 msgkey="query.required"
     */
    public void setQuery(String query) {
        this.query = query;
    }

    public String getQuery() {
        return query;
    }
}
```

32

How it maps

```
<form name="SearchForm">
```

```
<field property="query"
      depends="required">
```

```
<msg name="required"
      key="query.required"
/>
```

```
<arg0 key="SearchForm.query"/>
```

```
</field>
```

```
</form>
```

```
/**
 * Search query entry form.
 *
 * @struts.form      name="SearchForm"
 */
public class SearchForm extends ValidatorForm {
    private String query;

    /**
     * Sets the query attribute of the SearchForm object
     *
     * @struts.validator type="required"
     *                  msgkey="query.required"
     */
    public void setQuery(String query) {
        this.query = query;
    }

    public String getQuery() {
        return query;
    }
}
```

33

<webdoclet>

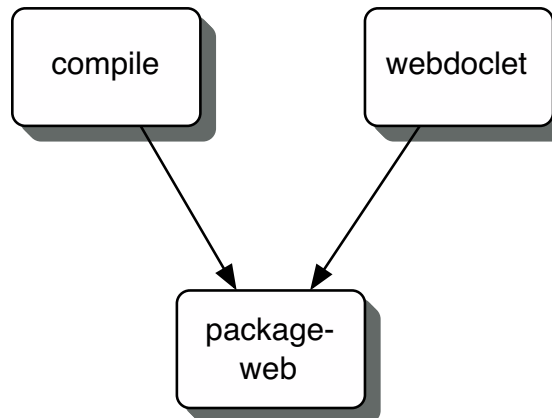
```
<webdoclet destdir="${build.dir}/web/WEB-INF"
           force="${xdoclet.force}"
           mergedir="metadata/web">

    <fileset dir="src/web"/>

    <deploymentdescriptor validatexml="true"
                        destdir="${build.dir}/${site}"
    />
    <jsptaglib validatexml="true"
              shortName="antbook"
              filename="antbook.tld"
    />
    <strutsconfigxml validatexml="true" version="1.1"/>
    <strutsvalidationxml omitdtd="true"/>
</webdoclet>
```

34

Web build process



35

Web build process

```
<target name="compile-web"
        depends="package-common,package-ejb">
    <javac...>
</target>

<target name="webdoclet" depends="init">
    <webdoclet...>
</target>

<target name="package-web"
        depends="init,compile-web,webdoclet">
    <jar...>
</target>
```

36

Future of XDoclet

- Velocity templating
- Jakarta project?
- Support for metadata JSR's
- XDoclet2 development in progress now
- And much more...

37

eXtreme preview

- Customizing templates
- Writing tag handlers
- Developing custom subtasks

38

web-security.xml

```
<XDtConfig:ifConfigParamEquals paramName="security" value="basic">
  <login-config>
    <auth-method>BASIC</auth-method>
  </login-config>
</XDtConfig:ifConfigParamEquals>

<XDtConfig:ifConfigParamEquals paramName="security" value="form">
  <login-config>
    <auth-method>FORM</auth-method>
    <realm-name>java:/jaas/myapp</realm-name>
    <form-login-config>

      <form-login-page>/login.jsp</form-login-page>

      <form-error-page>/login_fail.html</form-error-page>
    </form-login-config>
  </login-config>
</XDtConfig:ifConfigParamEquals>
```

39

Controlling security from build

```
<property name="web.security.mode" value="form"/>

<webdoclet...>
  .
  .
  .

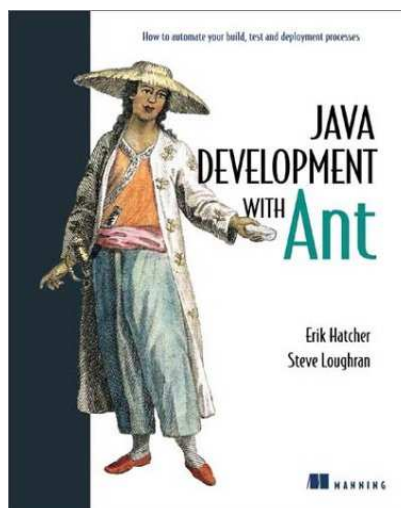
  <deploymentdescriptor validatexml="true" distributable="false">
    <configParam name="security" value="${web.security.mode}"/>
  </deploymentdescriptor>
</webdoclet>
```

40

Resources

- XDoclet
<http://xdoclet.sourceforge.net>
- JavaDevWithAnt
<http://www.ehatchersolutions.com/JavaDevWithAnt>
- Java Development with Ant
<http://www.manning.com/antbook>

41



42

