© Copyright 2003, Multitask Consulting Pty. Ltd.

Jelly: Executable XML

dIon Gillard

Multitask Consulting Pty. Ltd.

© Copyright 2003, Multitask Consulting Pty. Ltd.

Agenda

- What is Jelly?
- Anatomy of a Jelly script
- Jelly 'Core' Tag Library
- Other supplied tag libraries
- Scripting languages in Jelly
- Embedding Jelly

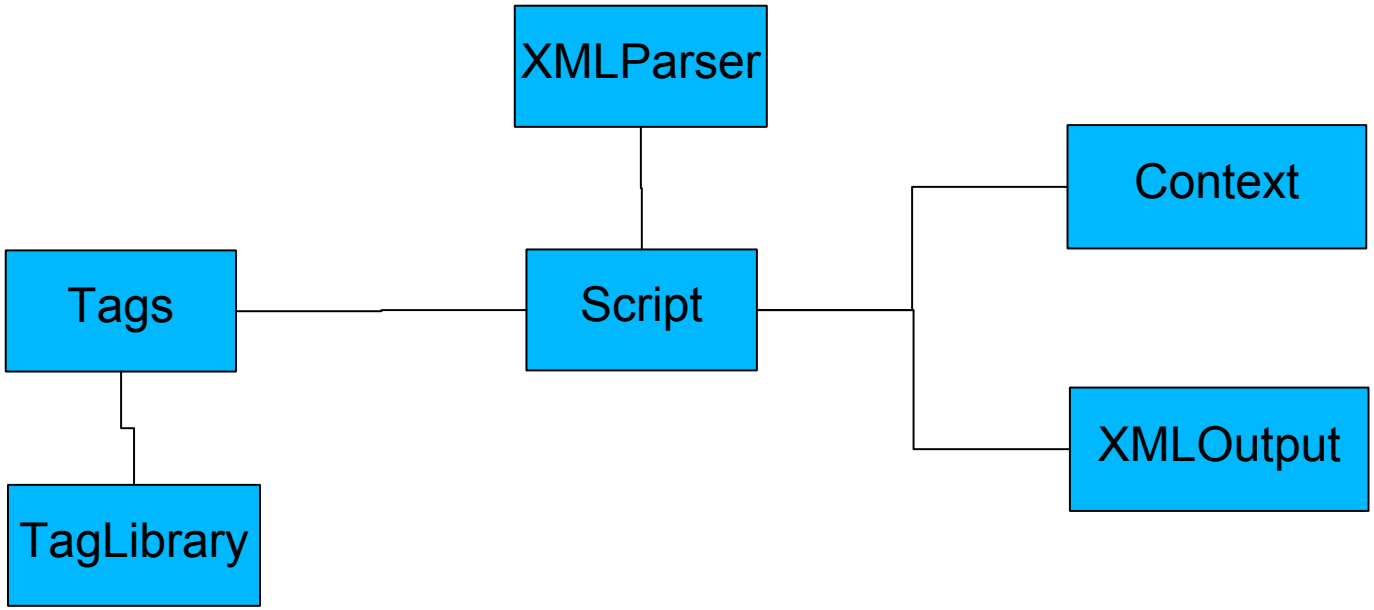
© Copyright 2003, Multitask Consulting Pty. Ltd.

What Is Jelly?

- Java and XML based scripting engine
- Executes XML and produces XML, HTML, *etc.*
- Front end to other XML tools such as Ant
- Borrows heavily from JSP and JSTL
- Comes with many tag libraries for common tasks
- Uses Jexl for expression support
- Extensible *via* Java or Jelly tag libraries

© Copyright 2003, Multitask Consulting Pty. Ltd.

Jelly Architecture



© Copyright 2003, Multitask Consulting Pty. Ltd.

Key classes

- XMLParser
  - Takes XML document and creates a script
- Context
  - Variables available to the Jelly script at run time
- Script
  - Executable 'code' made up of text and tags
- TagLibrary
  - Group of tags with a registered name

© Copyright 2003, Multitask Consulting Pty. Ltd.

Agenda

- What is Jelly?
- **Anatomy of a Jelly script**
- Jelly 'Core' Tag Library
- Other supplied tag libraries
- Scripting languages in Jelly
- Embedding Jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

A Jelly Script

```
<?xml version="1.0" ?>

<j:jelly xmlns:j="jelly:core">

  Hello World!

</j:jelly>
```

Demo: aJellyScript-1.jelly, aJellyScript-2.jelly, aJellyScript-3.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Jelly Script – Details

- It's an XML Document
- Namespaces used to refer to tag libraries
- Text is passed straight through
- Trimming? (Sample 1)
- Encoding? (Sample 2)
- jelly:core where is that defined?

© Copyright 2003, Multitask Consulting Pty. Ltd.

Jelly Script – variables

```
<?xml version="1.0" ?>
<jelly xmlns="jelly:core" trim="no">
  Properties
  ------------
  <forEach var="iter"
  items="${systemScope}">${iter.key} =
  ${iter.value}
  </forEach>
</jelly>
```

Demo: aJellyScript-4.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Jelly Script – variables *(Continued)*

- Default namespace
- forEach from core tag library
- systemScope predefined variable
- Accessing iterators and objects
- Text output formatting

© Copyright 2003, Multitask Consulting Pty. Ltd.

Agenda

- What is Jelly?
- Anatomy of a Jelly script
- **Jelly 'Core' Tag Library**
- Other supplied tag libraries
- Scripting languages in Jelly
- Embedding Jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core TagLibrary

- Jelly was originally one big jar with all the tag libraries included
- Too many dependencies were difficult to track
- Split into core and ant, antlr, avalon, bean, beanshell, betwixt, bsf, define, dynabean, email, fmt, html, http, interaction, jetty, jface, jms, jmx, jsl, junit, log, ojb, quartz, soap, sql, swing, swt, threads, util, validate, velocity, xml, xmlunit

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core TagLibrary *vs.* Others

- These were the tags that will most always be used
- We will not cover all tag libraries during the session
- If you become familiar with core, the documentation and a bit of reading between the lines will allow you to easily understand the others

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – set

- <j:set var="name" value="${expr}"/>
  - Or <j:set var="name">${stringExpr}</j:set>
- Assignment operator
- Not like Ant properties
- value can be any jexl expression

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – set Example

```
<?xml version="1.0" ?>
<jelly xmlns="jelly:core" trim="yes">
  Properties
  ------------
  <forEach var="iter" items="${systemScope}">
    <set var="name" value="${iter.key}"/>
    <set var="value" value="${iter.value}"/>
    ${name} = "${value}"
  </forEach>
</jelly>
```

Demo: core1.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – if

```
<if test="${booleanExpr}">
  ...more tags...
</if>
```

- Evaluates its body if the expression is true
- No else tag!
- operators supported >, >=, <, <=, !

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – if Example

```
<?xml version="1.0" ?>
<jelly xmlns="jelly:core" trim="no">

  <forEach var="iter" items="${systemScope}">
    <set var="name" value="${iter.key}"/>
    <if test="${name == 'user.name'}">
      Hi ${iter.value}!
    </if>
  </forEach>
</jelly>
```

Demo: core2.jelly, core3.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – forEach

- You've seen it in action
- The 'for loop' for jelly

```
<forEach var="item" items="${expr}"
  indexVar="counter" begin="number"
  end="number" step="number">
  ...statements...
</forEach>
```

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – new

- Create a new object
- <new var="name" className="fullName" />
- The class must be available to Jelly either in the current thread's context class loader, or on the classpath
- This is very different to Ant's <property/>
- Arguments can be passed to the constructor using <arg>

Demo: core4.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – new Example

```
<?xml version="1.0" ?>
<jelly xmlns="jelly:core" trim="no">

    <new var="myList"
      className="java.util.ArrayList" />
    Size is ${size(myList)}

</jelly>
```

Demo: core5.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – invoke and catch

- Invoke a method on an object
- Can take nested <arg>s like new

```
<invoke var="result" method="name"
        on="object"/>
```

- Exception handling?

```
<catch var="e">
  ...
</catch>
```

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – invoke Example

```
  <new var="myList1"
 className="java.util.ArrayList"/>
  <expr value="${myList1.add('Hello')}"/>
  <invoke var="dontCare" method="add"
          on="${myList1}">
    <arg value="World"/>
  </invoke>
  Size of list 1 is ${size(myList1)}
  list 1 is ${myList1}
```

Demo: core6.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – switch

- Switch, case, default from Java

```
<switch on="${expr}">
  <case value="${otherExpr}">
    ...
  </case>
  <default>
    ...
  </default>
</switch>
```

Demo: core8.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – switch Example

```
<set var="lookingFor" value="dion" />
<switch on="${systemScope['user.name']}">
  <case value="${lookingFor}">
    Gotcha
  </case>
  <default>
    Imposter!
  </default>
</switch>
```

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – choose

- Similar to switch, but not necessarily on a single value

```
<choose>
  <when test="${booleanExpr}">
    ...
  </when>
  <otherwise></otherwise>
</choose>
```

Demo: core9.jelly, core10.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – choose Example

```
<set var="lookingFor" value="dion" />
<choose>
  <when test="${systemScope['user.name'] ==
lookingFor}">
    Gotcha
  </when>
  <otherwise>
    Imposter!
  </otherwise>
</choose>
```

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – while

- While loop for Jelly
- <while test="${booleanExpr}">…</while>

```
<set var="start" value="5" />
<while test="${start != 10}">
  start is ${start}
  <set var="start" value="${start + 1}"/>
</while>
```

Demo: core11.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – break

- Breaks out of while or forEach
- <break test="${booleanExpr}">...</while>

```
<set var="start" value="5" />
<while test="${true}">
  start is ${start}
  <set var="start" value="${start + 1}"/>
  <break test="${start == 10}"/>
</while>
```

Demo: core12.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – thread

- Creates a new thread and runs its body on that thread
- Output can be to XMLOutput, file, or inherited
- All the usual threading issues apply
- Script finishes when the threads are completed

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – thread Example

```
<set var="start" value="5" />
<thread>
  <while test="${start != 10}">
    start is ${start}
    <set var="start" value="${start + 1}"/>
  </while>
  <new var="end" className="java.util.Date"/>
  Thread ended at ${end}
</thread>
<new var="now" className="java.util.Date"/>
Done ${now}
```

Demo: core13.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – useBean

- Create a bean of the given class
- Convert XML attributes into bean properties and call the setters
- Make it available as a variable
- <useBean var="name" class="fullName" property1="value" property2="value"/>

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – useBean Example

```
<new var="dimension"
     className="java.awt.Dimension">
  <arg type="int" value="320"/>
  <arg type="int" value="240"/>
</new>

<useBean var="frame"
class="javax.swing.JFrame" title="My Frame" />
  ${frame.setSize(dimension)}
<setProperties object="${frame}"
               visible="true"/>
```

Demo: core14.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – useList

- Create a new list and populate it from the given items
- <useList var="name" items="${expr}"/>

```
<useList var="myList"
  items="${systemScope.keySet()}" />
list is '${myList}'
```

Demo: core15.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – include and import

- Import or Include a script into the current one using a URI or file
- Include inherits values from the parent, and can export back if required
- Import by default doesn't inherit and can't export
- <import inherit="true/false" file="${name}" uri="${name}">
- <include export="true/false" ...>

Demo: importInclude.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core - file

- Write the tag body to a file, or variable

```
<file name="${fileName}"
  omitXmlDeclaration="${bool}"
  outputMode="XML/HTML" prettyPrint="${bool}"
  encoding="${value}" var="alternateVariable"
  escapeText="${bool}">
   ...
</file>
```

Demo: core16.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – file Example

```
<file name="${systemScope['user.home']}
   /temp.jelly" omitXmlDeclaration="true">
  When you hear the air attack warning...
 </file>
```

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – remove, expr, scope

- Remove
  - Removes a variable from the current scope
  - <remove var="name"/>
- Expr
  - Evaluates an expression
  - <expr value="${someExpression}"/>
- Scope
  - Sets up a new scope that won't affect current variables

Demo: scope.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Core – jelly and whitespace

- <jelly>…</jelly>
  - Runs it's body
- <whitespace trim="${bool}"> …</whitespace>
  - Container to allow fine grained control over whitespace

© Copyright 2003, Multitask Consulting Pty. Ltd.

Agenda

- What is Jelly?
- Anatomy of a Jelly script
- Jelly 'Core' Tag Library
- **Other supplied tag libraries**
- Scripting languages in Jelly
- Embedding Jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Other TagLibraries

- Ant – allows ant tasks to be invoked from Jelly
- Antlr – Generate code from antlr grammars
- Avalon – An Avalon-based service to run scripts
- Bean – map tags to beans like Ant does
- BeanShell – runs beanshell scripts
- Betwixt – turn beans into XML and XML into beans

© Copyright 2003, Multitask Consulting Pty. Ltd.

Other TagLibraries *(Continued)*

- BSF – run any BSF capable scripting language from Jelly
- Define – define new tag libraries and tags *via* jelly tags
- DynaBean – tags to define, create and manipulate dyna beans
- Email – send emails
- Fmt – jstl format tag library

© Copyright 2003, Multitask Consulting Pty. Ltd.

Other TagLibraries *(Continued)*

- Html – parse possibly badly formatted HTML as XML
- Http – perform common HTTP protocol actions
- Interaction – prompt the user and get replies
- Jetty – manipulate a jetty instance, process requests *via* Jelly
- JMS – work with messages
- JSL – simplified stylesheet processing using objects

© Copyright 2003, Multitask Consulting Pty. Ltd.

Other TagLibraries *(Continued)*

- Junit – write unit tests as jelly scripts
- Log – wrapper for commons logging
- Ojb – store objects in OJB
- Quartz – cron like scheduling *via* scripts
- Soap – Invoke a web service
- SQL – JSTL. Create transactions, run queries, process result sets, update data
- Swing – build UI *via* tags

© Copyright 2003, Multitask Consulting Pty. Ltd.

Other TagLibraries *(Continued)*

- Swt – build eclipse style UI *via* tags
- Threads – coordinate across threads
- Util – miscellaneous extras
- Validate – validate XML docs
- Velocity – run velocity templates in Jelly
- Xml – parse, sort, use xpath for if, transform and loop through XML docs
- Xmlunit – unit test XML document content

© Copyright 2003, Multitask Consulting Pty. Ltd.

Ant Tag Library

- Any ant task typedef can be used except those that rely on build.xml
- Use xmlns:ant="jelly:ant"
- Place the commons-jelly-tags-ant-XXX.jar in the classpath, along with commons-grant.jar, ant.jar and ant-optional.jar
- Tags provided: ant, fileScanner, setProperty

© Copyright 2003, Multitask Consulting Pty. Ltd.

Ant Tag Library *(Continued)*

```
<fileScanner var="scripts">
  <fileset dir="C:/source/css/css2003">
    <include name="*.jelly"/>
  </fileset>
</fileScanner>

<j:forEach var="script"
 items="${scripts.iterator()}" trim="yes">
  <echo>script is ${script}</echo>
</j:forEach>
```

Demo: ant1.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Ant Tag Library – setProperty

- Sets a property on an Ant task if the value is not null.

```
<ant:javadoc....>
  <!-- allow custom doclets -->
  <j:if test="${context.getVariable('maven.javadoc.
  doclet') != null}">
    <ant:setProperty name="doclet"
      value="${maven.javadoc.doclet}" />
  </j:if>
</ant:javadoc>
```

© Copyright 2003, Multitask Consulting Pty. Ltd.

XML Taglib

- Parse
  - <x:parse var="name" xml="${uriFileReaderOrInputStream}"/>
  - ${name} is now a Dom4J Document
  - xmlns:x="jelly:xml"
  - Need taglib jar available
- Other Sessions:
  - Bonnie B. Ricca "XSLT 2.0: Not Your Mothers XSLT"

© Copyright 2003, Multitask Consulting Pty. Ltd.

XML Taglib – parse Example

```
<x:parse var="doc">
 <document>
   <properties><title>Doc Title</title></properties>
   <body>
     <section>Hello World</section>
   </body>
 </document>
</x:parse>
Title was '<x:expr  select="$doc/document/properties/title"/>'
Section was '<x:expr select="$doc/document/body/section"/>'
```

Demo: xml1.jelly, xml2.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

XML Taglib – if Example

```
<x:if select="$doc/document/body/section[@name =
  'fred']">
  found section fred
</x:if>
```

- Allows logic to use xpath queries on an XML document to drive processing

Demo: xml3.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

XML Taglib – forEach Example

```
<x:forEach var="name" select="$doc/document/body/section/@name"
  trim="no">
  Section name is ${name.text}
</x:forEach>
```

- Loop through XML documents processing nodes as needed.
- Similar to forEach, except nodeset comes from xpath

Demo: xml5.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

XML Taglib – sort Example

```
<x:set var="sections" select="$doc/document/body/section" />
<x:sort list="${sections}" sort="./@name" descending="false" />
<j:forEach var="name" items="${sections}">
   Section name is ${name.text}
</j:forEach>
```

- Data can be easily sorted before output

Demo: xml4.jelly, xml6.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

XML Taglib – set Example

```
<x:set var="name"
  select="string($doc/document/body/section/@name)"/>
Name is ${name}
```

- Variables can be set based on the result of an xpath expression

Demo: xml4.jelly, xml7.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

SQL Taglib

- Allows processing of SQL queries
- Transformation into XML/HTML easily
- Based on the JSTL taglib
- Tags provided
  - Transaction, driver, query, update, resultSet, row, dataSource

© Copyright 2003, Multitask Consulting Pty. Ltd.

SQL Taglib – Example

```
<sql:setDataSource
  url="jdbc:db2:SAMPLE"
  driver="COM.ibm.db2...."
  user="db2admin"
  password="db2admin"/>

<sql:query var="results">
    select * from ${databaseTable}
</sql:query>
```

Demo: sql1.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

SQL Taglib – Example *(Continued)*

```
<j:forEach items="${results.rowsByIndex}"
           var="row">
  <j:forEach var="columnName"
  items="${results.columnNames}" indexVar="i">
    ${columnName} = ${row[i]}
  </j:forEach>

</j:forEach>
```

Demo: sql2.jelly, sql3.jelly, sql4.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Util Taglib

- Miscellaneous utility tags
- <util:available file="${fileName}" uri="relativePath">...</util:available>
- <util:loadText var="name" file="${fileName}" uri="relativePath" />
- <util:tokenize var="name" delim="${char}">${stringContent}</util:tokenize>

Demo: util1.jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Util Taglib *(Continued)*

- <util:file var="name" name="${fileName}"/>
- <util:properties var="name" file="${fileName}" uri="relativePath" />
- <util:replace var="name" oldChar="\" newChar="/" value="${someString}"/>

© Copyright 2003, Multitask Consulting Pty. Ltd.

User Interfaces

- Jelly comes with both a Swing and SWT tag library
- This allows the UI to be specified in XML
- It's not XUL

Demo: swing/ maven demo:swing, swt/ maven demo

© Copyright 2003, Multitask Consulting Pty. Ltd.

Agenda

- What is Jelly?
- Anatomy of a Jelly script
- Jelly 'Core' Tag Library
- Other supplied tag libraries
- **Scripting languages in Jelly**
- Embedding Jelly

© Copyright 2003, Multitask Consulting Pty. Ltd.

Scripting Languages

- XML is often unwieldy
- Code can be far more compact and expressive, but is difficult to generate
- As a scripting language itself, Jelly is unusual in that it lets you load and run scripts for other languages within a script

© Copyright 2003, Multitask Consulting Pty. Ltd.

Scripting – BeanShell

- BeanShell tag library
- Need bean shell jars
- Use xmlns:bsh="jelly:beanshell"

```
<bsh:script>
  Properties sysprops =
System.getProperties();
  System.out.println("-testsysprop =
" + sysprops.get("testsysprop"));
</bsh:script>
```

© Copyright 2003, Multitask Consulting Pty. Ltd.

Scripting – BSF

- Ex-IBM Bean Scripting Framework
- Need BSF jars
- Use xmlns:javascript="jelly:javascript”
- Provides Jython, Pnuts, JavaScript
- Registers Jelly Context with BSF Manager
- Expressions are passed through to BSF to manage

© Copyright 2003, Multitask Consulting Pty. Ltd.

Scripting – BSF *(Continued)*

- JavaScript samples are currently broken in Jelly

```
<j:jelly xmlns="jelly:javascript" xmlns:j="jelly:core"
  trim="no">
<script>
 var result = 0
 for (var i = 0; i != 10; i++) {
  result  += i;
 }
</script>
```

© Copyright 2003, Multitask Consulting Pty. Ltd.

Agenda

- What is Jelly?
- Anatomy of a Jelly script
- Jelly 'Core' Tag Library
- Other supplied tag libraries
- Scripting languages in Jelly
- **Embedding Jelly**

© Copyright 2003, Multitask Consulting Pty. Ltd.

Embedding Jelly

- Where is the output to go?
- What variables do you want available to the script?
- Set up XMLOutput
- Run the script
- What do you want to do with the output?

© Copyright 2003, Multitask Consulting Pty. Ltd.

Embedding Jelly – Sample

```
OutputStream output = new FileOutputStream("output.xml");
JellyContext context = new JellyContext();
// populate it
context.setVariable("bean1", myApp.getData());
// create XMLOutput
XMLOutput xmlOutput = XMLOutput.createXMLOutput(output);
// run the script specified
context.runScript(scriptFileName, xmlOutput);
xmlOutput.flush();
output.close();
```

© Copyright 2003, Multitask Consulting Pty. Ltd.

Embedding Jelly – Misc

- Need Jelly Core as a minimum dependency
- Other taglibs as required
- Running scripts gives users more control
- They will do things you didn't plan on
- Jelly is good at swallowing exceptions :-(
- What do you do with the output
- How do you tell the user of script results, *etc.*

© Copyright 2003, Multitask Consulting Pty. Ltd.

Summary

- XML based scripting
- Large tag library support
- Based on JSTL / JSP
- XML can be intermixed with Java and tags to produce output

© Copyright 2003, Multitask Consulting Pty. Ltd.

Resources

- http://jakarta.apache.org/commons/jelly/
- http://jakarta.apache.org/commons/jexl/
- http://cocoon.apache.org/
- http://maven.apache.org/
- dion@multitask.com.au