Lotusphere Europe

AD109 - Java Programming with Domino: A Look at Servlets and JSPs

Becky Gibson
Iris Associates

Lotus.

WORK AS ONE

Agenda

- Using Servlets in Domino
- JSP Basics
- Using JSPs with Domino
- Demo of Servlets, JSPs
- Deployment Considerations

What is a Servlet?

- Java Object that extends a server
  - written to a standard interface - Java Servlet API
- Serves up custom data
- Gateway object
  - between web server and resident objects
- Message handler
  - Requests In
  - Responses Out

Servlet Support in Domino R5

- Built in Java Servlet Engine for JSDK 2.0
- Servlet Manager is pluggable
- IBM WebSphere 2.0, 3.0 compatible

Configuring Domino for Servlets

- Server Record
  - Internet Protocols Tab
    - Domino Web Engine Tab
    - Java Servlets Section

Java Servlets	
Java servlet support:	**Domino Servlet Manager**
Servlet URL path:	/servlet
Class path:	domino/servlet
Servlet file extensions:	
Session state tracking:	Enabled
Idle session time-out:	30 minutes
Maximum active sessions:	1024
Session persistence:	Disabled

Tools for Developing a Servlet

- JSDK 2.0 documentation at java.sun.com
- Commercial Java Development Packages
  - VisualAge 2.0
- No Frills route
  - put JSDK.jar (packaged with R5) into classpath
  - javac

Writing the Servlet

- Extend javax.servlet.http.HttpServlet
- Implement the desired methods
  - doGet()
  - doPost()
  - init() & destroy()
- place class files into servlet dir
  - <Domino data dir>/domino/servlet
- Invoke by URL
  - /servlet/name
  - http://bgibsontpad/servlet/Hello?userName=Becky

Simple Example

```
import javax.servlet.*;
import javax.servlet.http.*;
import lotus.domino.*;
import java.io.*;

public class Hello extends HttpServlet {
        public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
                res.setContentType("text/html");
                PrintWriter out = res.getWriter();
                String user = req.getParameter("userName");
                try {
                        NotesThread.sinitThread();
                        Session session = NotesFactory.createSession();
                        String serverName = session.getServerName();
                        String userName = session.getUserName();
                        StringBuffer s = new StringBuffer();
                        s.append("<HTML><HEAD><TITLE>Welcome to Lotusphere Europe  2000</TITLE><BODY><BR><BR><BR>");
                        s.append("<CENTER><FONT FACE=\"Comic Sans MS\" SIZE=\"7\">Welcome to Lotusphere Europe 2000");
                        if (user != null && user.length() > 0)
                                s.append(", " + user);
                        s.append("!</FONT><BR><FONT SIZE=\"5\">This servlet is running on : " + serverName + "<BR>");
                        s.append("The user name is: " + userName + "</FONT><BR></CENTER></BODY></HTML>");
                        out.println(s.toString());
                        out.close();
                        session.recycle(); // clean up objects
                } catch (Exception e) {
                        e.printStackTrace();
                }       finally {
                        NotesThread.stermThread();
                }
        } // end of doGet
} // end of Hello class
```

Lotusphere Europe

Servlet Startup

- Servlet Manager ClassLoader loads servlet
  - uses domino/servlet path
- System ClassLoader loads other classes
  - locates using CLASSPATH
- init() method executes
- doGet() or doPost() executes for each request
- Servlet classes REUSED

Debugging the Servlet

- Must reload after each change
  - tell http quit & load http
  - tell http restart
- System.out.println("messages display on the server console");

Servlet Security Issues

- Servlet code is "trusted"
- Security of place
- R4.6/R5 runs under server authority
- As of R5.02 - ability to establish user authority
  - enabled within servlet code
  - NotesFactory.createSession("",username,password);

Maintaining the Session State

- Servlet support for shared, persistent state
- Implement HttpSession Interface
- Maintained across HTTP requests and restarts
  - across restarts only if implement java.io.serialisable

Servlet Configuration via servlets.properties File

- <Domino data dir>\servlets.properties
- Replaces servlet.cnf (R4.6)
- Standard Java Properties file format
- Directives:
  - servlets.startup=<name1> <name2>
  - servlet.<name>.initArgs=<argname=value>,<arg2=value>
  - servlet.alias.code=<class>
  - servlet.<name>.extension=<ext> <ext2>

Servlet Summary

- Extend server capabilities
- Performance is critical
- Need to maintain session state
- Want full control over HTML generation

What is a JSP?

- JavaServer Page
  - JavaServer Pages Specification from java.sun.com
- Java based technology for developing dynamic web pages
  - separates the presentation of data from logic
- New type of web 'page'
  - Java code embedded in the page with HTML markup
  - Additional functionality accessed via a set of markup tags

How do JavaServer Pages Work?

- Based on Servlets
  - access to normal HTTP Request parameters
- JSP page gets compiled into servlet
- Servlet returns resulting HTML
- As long as page doesn't change, each request for the page executes the existing servlet

Using JSPs with Domino

- Domino does not provide integrated JSP support
- Use a third party JSP container
  - Tomcat, WebSphere
- Install JSP container
- Run Domino & JSP container
- For JSP support access JSP container directly
  - http://bgibsontpad:8080/Hello.jsp?userName=Becky

Simple JSP

```
<HTML>
<HEAD>
<TITLE>Welcome to Lotusphere Europe</TITLE>
</HEAD><BODY>
<% String name=request.getParameter("userName");
        if (name == null) name = "Devoted Notes Developer";
%>
<BR><BR><BR><CENTER><FONT FACE="Comic Sans MS" SIZE="7">
Welcome to Lotusphere Europe 2000, <%= name %><BR>
<%@ page import="javax.servlet.*,  javax.servlet.http.*, java.io.*, lotus.domino.*" %>
<% String serverName = "";
        String userName = "";
        try {
                        NotesThread.sinitThread();
                        Session theSession = NotesFactory.createSession();
                        serverName = theSession.getServerName();
                        userName = theSession.getUserName();
        } catch (Exception e) {
                e.printStackTrace();
        } finally {
                NotesThread.stermThread();
        }
%>
<FONT SIZE=5>The server name is: <%= serverName %><BR>
The user name is: <%= userName %><BR></FONT></CENTER>
</BODY></HTML>
```

Lotusphere Europe

2004

JSP Tags

- Four categories of tags
  - directives
  - scripting elements
  - comments
  - actions
- General syntax
  - <%x tagname attribute=value %>
- All tags also support XML-based syntax
  - <jsp:type.name attribute=value />

Tag Examples

- Directives
  - <%@ page language="java" %>
  - <%@ include file="relativeURL" %>
- Scripting Tags
  - declaration <%! int x=0 %>
  - expression <%= object.property %>
  - code fragment<% scriptlet %>
- Actions
  - <jsp:useBean id=beanName class="package.class" />
  - <jsp:getProperty name=beanName property="name" />

Using JSP with Domino Objects

- Create a Bean to access Domino Objects
  - <jsp:useBean id="pReader" class="ProfileReader">
  - <jsp:setProperty name="pReader" property="docID" param="docID" />
  - </jsp:useBean>
- Use the Bean on the JSP page to interact with Domino data
  - <% String title = pReader.getDocTitle(); %>
  - <B>Title is <%= title %></B>

Lotusphere Europe 2004

JSP Tag Libraries

- Introduced in JSP 1.1 spec
- Allows you to create custom tags
  - Wrap common behavior into HTML like tag
- Written to the JSP Tag Extension API
  - package javax.servlet.jsp.tagext.*
- Defined in Tag Library Descriptor
  - XML file that describes the tag, parameters, & class files

Using a Tag Library

- Specified via a Directive JSP tag
  - <%@ taglib uri="tagLibraryURI" prefix="Domino" %>
- uri - how to locate the tag library
- prefix - namespace assigned to the library
- example:
  - <%@ taglib uri="/WEB-INF/mylib.tld"prefix="util"%>
  - Current time is < util:date />

Possible Domino Tags

- Session
- Database
- Document
- View
- Item
- Query
- Loop
- Formula

Simple Tag Example

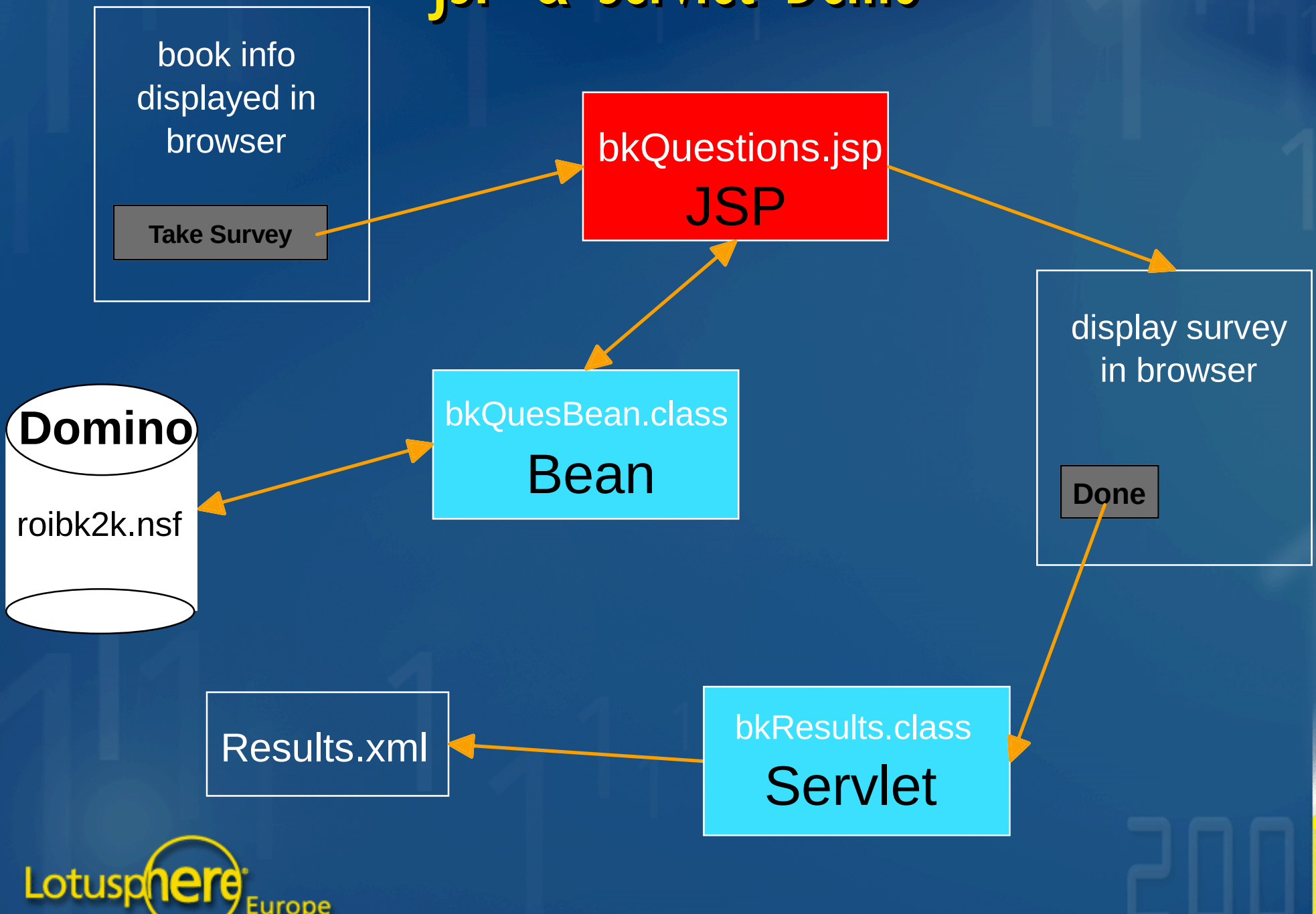
```
<HTML>
<HEAD>
<TITLE>Welcome to Lotusphere Europe 2000  - Tag Library Version</TITLE>
</HEAD><BODY>
<%@ taglib uri="domino.tld" prefix="Domino" %>
<%@ page language="java" import="lotus.domino.*" %>
<% String name=request.getParameter("userName");
        if (name == null) name = "Devoted Notes Developer";
%>
<BR><BR><BR><CENTER><FONT FACE="Comic Sans MS" SIZE="7">
Welcome to Lotusphere Europe 2000, <%= name %><BR>

<% String serverName = "";
        String userName = "";
%>
<Domino:session id="sInfo">
<%    Session theSession = sInfo.getSession();
                serverName = theSession.getServerName();
                userName = theSession.getUserName();
%>
</Domino:session>
<H1> Tag Library Example</H1>
<FONT SIZE=5>The server name is: <%= serverName %><BR>
The user name is: <%= userName %><BR></FONT></CENTER>
</BODY></HTML>
```

Lotusphere Europe

JSP & Servlet Demo

JSP & Servlet Demo



Deployment Issues

- Servlets
  - Extend web server capabilities
  - Software runs at one location
  - Remember state between requests
- JSP
  - Need dynamic page content
  - Separate UI from page logic

Futures

- Next Major Release of Domino
  - Servlet 2.2
  - JSP 1.1
  - LotusScript support in JSPs
  - XML/XSLT Support
  - Domino Tag Libraries
  - Designer Support

Summary

- Extend server capabilities with Servlets & JSPs
- Servlet 2.0 integrated into Domino
- JSP support possible today via 3rd party container
- Access to Domino Objects
- Integrate both technologies into your Domino web apps

Questions?

Please complete your evaluations

Lotusphere Europe

2004

References

- Servlets
  - http://www.javasoft.com/products/servlet/index.html
  - http://jakarta.apache.org/tomcat
  - http://www.servletcentral.com
- JSP
  - http://www.javasoft.com/products/jsp/index.html
  - http://java.sun.com/products/jsp/technical.html#syntax
- JSP Tag Libraries
  - http://www.javasoft.com/products/jsp/taglibraries.html
  - http://jsptags.com/