CF and JSP/Servlets

Developed originally by Robi Sen
For the CF UnderGround II Seminar, Apr 2001
Edited and enhanced by Charlie Arehart
(Robi had an emergency and couldn’t make it)

Topics Covered

- Overview JSP/Servlets
- Comparison of CF and JSP Tags
  - Variables
  - Application Scope
  - Conditional Processing
  - Reusing Common Code
  - Sessions
  - Database Access
- JSP/Servlet Engine, Tool Vendors
- Leveraging Java in CF Today
- Learning More

What are they?

- JSP/Java Servlets
  - Servlets
    - Comparable to CGI/ISAPI/NSAPI programming, in Java vs Perl
    - Java class that dynamically extends the function of a Web Server
    - Handle HTTP requests and generate HTTP responses
  - JSP
    - Tag-based scripting and page-template interface to Java development, a la CF
    - High level abstraction language to servlet standard
Server Process for JSP

What CF has over JSP

- ColdFusion offers:
  - Faster learning curve
  - More features in language out of the box
    - (i.e. cftransaction, cfhttp,cfftp, cached queries, queries of queries)
  - Greater abstraction, high productivity
  - Greater maturity as web application

What JSP over CF

- JSP offers:
  - Platform Agnostic (Write Once Run Anywhere)
  - Scalability and Robustness
  - Performance and power
  - Access to Enterprise Technologies
  - Manageability
  - Standardization
  - Massive adoption and developer community
JSP VS CF

- As well as:
  - Greater acceptance
  - Better reputation
  - More developer resources
  - And being based on Java
    - Object oriented
    - Many libraries

Underlying JSP/Servlets is Java

- Not really appropriate to see JSP as “just an alternative scripting environment”
  - Yes, pretty easy to compare simple things
  - Really need to understand Java to use effectively
    - And to fully leverage the power it brings
- Underlying JSP is servlets
  - Some things easier to do in one or the other
  - JSP generally favored when creating lots of HTML on a page
  - JSP can be seen as your entrée to servlets

Exploring JSP vs. CF Tags

- CF
  - Begin with CF (e.g., `<CFOUTPUT>`)%
  - Most have closing tags (e.g., `<CFOUTPUT>HTML code</CFOUTPUT>`)
- JSP
  - Begin with `<% and end with `%`
  - Contain Java code, expressions, directives, etc.
CF Tags vs. JSP Tags

- May help to consider comparing CF and JSP for performing common tasks
  - `<CFSET>` ≅ `<%! %>`
  - `<CFSCRIPT>` ≅ `<% %>`
  - `<CFOUTPUT>` ≅ `<%= %>`
  - `<%@ %>`
    - `<CFCONTENT>` (set the output MIME type) vs `<%@ page contentType="text/xml" %>`
    - `<CFAPPLICATION>` (turn on session-state management)

Where Files Are Stored

- Will depend on Java App Server
  - I’m using Jrun, which supports multiple servers, and multiple applications—doing demo in “Demo” server
    - Files stored at:
      - `D:\Program Files\Allaire\JRun\servers\default\demo-app.jsp`
    - JRUN sets up web server mapping to find files at:
      - `http://localhost:8100/demo/jsp/filename.jsp`
    - Have set up mapping in Studio to enable browse

Variables

- **Variable Type (string, integer, etc.)**
  - Type-less in ColdFusion
  - Strongly typed in JAVA
- **Case Sensitivity**
  - Ignored in CF
  - Enforced in JSP
Defining Variables

- **CF**
  ```
  <CFSET firstName="John">
  <CFOUTPUT>Hello #firstName#</CFOUTPUT>
  ```

- **JSP**
  ```
  <%= String firstName = "John"; %>
  Hello <%= firstName %>
  ```

Can also perform “pure” java statements within JSP, as a “scriptlet”
- which can be useful in some situations
  - though not particularly so, here

  ```
  <%= String fName = "John";
  out.println("Hello " + fName);
  ```

Conditional Processing

- `<CFIF>`
  ```
  <CFIF expression>
  HTML and CFML tags executed if expression is true
  </CFELSE>
  HTML and CFML tags executed if expression is false
  </CFIF>
  ```

- `if/else in pure Java (servlet, class, scriptlet)`
  ```
  if(expression) {
  // Java code to execute if expression is true
  } else {
  // Java code to execute if expression is false
  }
  ```
Conditional Processing

- if/else in JSP
  
  ```
  <% if(expression) { %>
      HTML and JSP tags executed if expression is true
  <% } else { %>
      HTML and JSP tags executed if expression is false
  <% } %>
  ```

Conditional Processing

- Conditional Expressions in CF/JSP
  - Really about CF vs Java expressions, as in:
    - `IS` vs `==` or `.equals()`
    - `IS NOT` vs `!=`

```
<CFLOOP>
  <CFLOOP FROM="1" TO="10" INDEX="i">'
  <CFOUTPUT>#i#</CFOUTPUT><BR>
</CFLOOP>
```

- "for" loop in pure Java
  ```
  <% for(int i=1; i<=10; i++) {%
      out.println(i + "<BR>");
  %} ```
Conditional Processing

- "for" loop in JSP
  ```
  <% for(int i=1; i<=10; i++) { %>
  <%= i %> <BR>
  <% } %>
  ```

Reusing Common Code

- CF
  ```
  <CFINCLUDE TEMPLATE="/Templates/header.cfm">
  ```
- JSP
  ```
  <% include file = "path" ... %>
  ```
- or
  ```
  <jsp:include page="/Templates/header.jsp"/>
  ```
- More like CF custom tag call
  ```
  # Goes to other page, executes, and returns
  # Passes request object to called page
  ```

Redirection

- CF
  ```
  <CFLOCATION URL="/Forms/demo.cfm">
  ```
- Java
  ```
  RequestDispatcher aDispatcher =
  request.getRequestDispatcher("/Templates/head er.jsp");
  aDispatcher.include(request, response); %>
  ```
- JSP
  ```
  <jsp:forward page="/Forms/demo.cfm" />
  ```
Comments

- CF
  - <!-- comment -->

- Java
  - <\* // one line comment;
  - \* multi
  - line comment */ %>

- JSP
  - <%= comment %>

Session State Maintenance

- CF
  - Cookies
  - Application.cfm
  - Application variables
  - Session variables

Application Scope

- Application Variable
  - Shared among all users of application
  - <CFSET Application.myVariable="somevalue">\n  - #Application.myVariable#

- Application object in JSP
  - Shared among all users of application
  - <% application.setAttribute("myVariable", "somevalue");
  - out.println(application.getAttribute("myVariable")
  - }); %>
Application Scope

- ServletContext object
  - Shared among all users of servlet
  ```java
  getServletContext().setAttribute("myVariable", "somevalue");
  getServletContext().getAttribute("myVariable");
  ```

Session State Maintenance

- CF "session." variables
  ```cf
  <cfset session.name = "john doe">
  ```
- Servlet HttpSession object
  ```cf
  <cfset aSession = request.getSession();
  aSession.setAttribute("name", "John Doe");
  ```
- JSP session object is an instance of the HttpSession object.
  ```cf
  <cfset session.setAttribute("name", "John Doe");
  ```

Database Access

- ODBC
  - Standard database access method
  - Inserts a middle layer (driver) between the database and the application
- JDBC (Java Database Connectivity)
  - Based on ODBC
  - Allows access to any tabular data source from the Java programming language
**Database Access: In CF**

- Use CF Administrator to set the DataSource
- Query the database using `<CFQUERY>`
  ```
  <CFSET variables.anID = 2>
  <CFQUERY NAME="myquery" DATASOURCE="mydatabase">
  select firstname, lastname from mytable
  where id = #variables.anID#</CFQUERY>
  ```
- Accessing the data from the ResultSet
  ```
  #myquery.firstname#
  #myquery.lastname#
  ```

**Database Access: In CF**

- Displaying the ResultSet
  - One Row
    ```
    <CFOUTPUT>
    The name is #myquery.firstname#
    #myquery.lastname#
    </CFOUTPUT>
    ```
  - Many Rows
    ```
    <CFOUTPUT QUERY="myquery">
    The name is #myquery.firstname#
    #myquery.lastname#
    </CFOUTPUT>
    ```

**Database Access: In Java**

- Set the DataSource using a GUI tool (e.g., Jrun Mgt Console)
  - In “default server”
  - Edit “jdbc data sources”
  - Click edit to create a new one
  - If already defined on server in odbc
    - Enter its name, in “name” (e.g., cfexamples)
    - Enter sun.jdbc.odbc.JdbcOdbcDriver for “driver”
    - Enter “jdbc:odbc:cfexamples” for url
    - Enter any other needed info (userid, password)
  - “Update”, then “test”
Database Access: In Java

- In page, import needed libraries
  
  ```
  <%@ page import="java.sql.*, javax.sql.*,
  javax.naming.*" %>
  ```

- Obtain a reference to the DataSource using JNDI
  
  ```
  InitialContext aContext = new InitialContext();
  DataSource myDataSource = (DataSource)
  aContext.lookup("java:comp/env/jdbc/cfexamples");
  ```

Database Access: In Java

- Call the DataSource method
  getConnection() to establish a connection
  
  ```
  Connection con =
  myDataSource.getConnection();
  ```

Database Access: In Java

- Create/Prepare the Statement
  
  ```
  PreparedStatement aStatement =
  con.prepareStatement("select firstname,
  lastname from cfexamples where empid = ?");
  aStatement.setInt(1, 2);
  ```

- Sets the first parameter (?) to the value 2
  - Finds empid=2
Database Access: In Java

- Execute the query using the Statement object's method executeQuery() method or the CallableStatement object's execute() method.
  
  ```java
  ResultSet rs = aStatement.executeQuery();
  ```

- Accessing the data from the ResultSet
  
  ```java
  rs.getString(1);
  rs.getString(2);
  ```

Displaying ResultSet: Scriptlets

- One Row
  
  ```java
  if(rs.next()) {
    out.println("Hello " +
    rs.getString(1) + 
    " " +
    rs.getString(2));
  }
  ```

- Many Rows
  
  ```java
  while(rs.next()) {
    out.println("Hello " +
                rs.getString(1) + 
                " " +
                rs.getString(2));
  }
  ```

Displaying ResultSet: JSP

- One Row
  
  ```jsp
  <% if(rs.next()) { %>
    Hello <%= rs.getString(1); %>
    <%= rs.getString(2); %>
  <% } %>
  ```

- Many Rows
  
  ```jsp
  <% while(rs.next()) { %>
    Hello <%= rs.getString(1); %>
    <%= rs.getString(2); %>
  <% } %>
  ```
More topics to learn

- Java
  - Language, libraries, data types
  - Concepts like classes, methods, packages, public/private/protected/"friendly", static/final, much more
- J2EE
  - JDBC, Enterprise Java Beans, JINI, JNDI, JMS, etc.
- JSP
  - JSP Custom Tags
  - JSP Page Directives
  - Error Handling …

More topics to learn

- SQL in scripts vs EJB
- Servlets/JSP
  - Request/response objects, headers, response codes
- Integrating servlets and JSP’s
  - Battle line among supporters of each
  - Best used in tandem, where each best fits
- And much more

Learning More

- Excellent documentation with Jrun
- Several books
  - Core Servlets and JSP, Marty Hall
  - Professional JSP, Wrox Press
  - Pure JSP, James Goodwill
  - Java Server Pages Application Development, Scott Stirling, Ben Forta, et al
  - And others
- Thinking in Java, Bruce Eckel
  - eckelobjects.com
Learning More

- Several CFDJ Articles
  - Java For Cfers, Ben Forta
    - 3 parts, starting November 2000
  - ColdFusion & Java: A Cold Cup o’ Joe, Guy Rish
    - 9 parts, starting in Jan 2001
  - Also see Java Developer’s Journal

JSP/Servlet Engine Providers

- Allaire Jrun
  - 3 person developer edition available free!
  - Can install on same server as CF Server
- IBM WebSphere
- BEA WebLogic
- Apache/TomCat
- others

Java Editing Tools

- Jrun Studio
  - CF Studio also supports JSP
- Kawa
- Others, from competing JSP engine providers
CF 6.0 AKA NEO

- CF, as we know it
  - But on top of a Java, rather than C++ platform
  - Basically transparent to CF developers
- Strength of JAVA, ease of CF
  - Backwards compatibility
  - Scalability built on a leading container (JRUN)
- May be made available on other Java Server vendor platforms (IBM, BEA, etc.)
  - Still being debated by Allaire, I understand

Leveraging Java in CF Today

- CFServlet
- CFOBJECT
- Java Custom Tags
- TagServlet (from n-ary.com)
- "wolf in sheep's clothing" trick
  - How to look like you're converting your CF code to use JSP when you're really not 😊

Times Up!

- Hope you enjoyed the session
- Send questions to:
  - Charlie Arehart
  - Carehart@systemanage.com