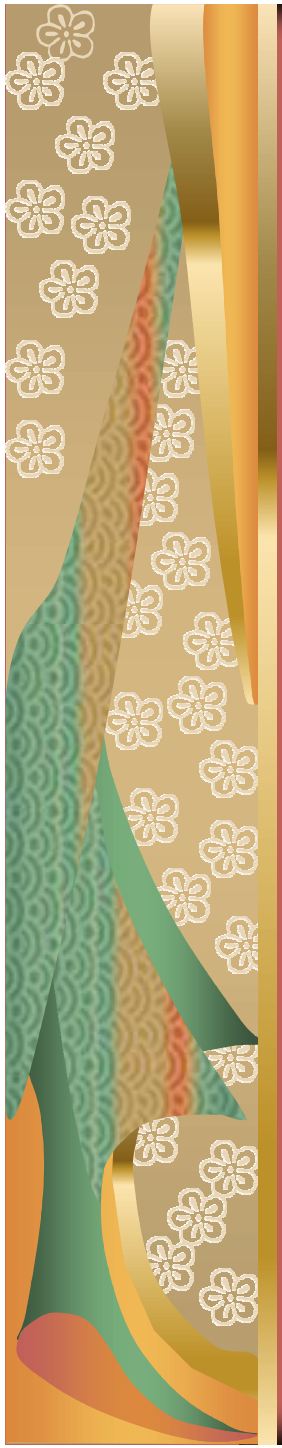
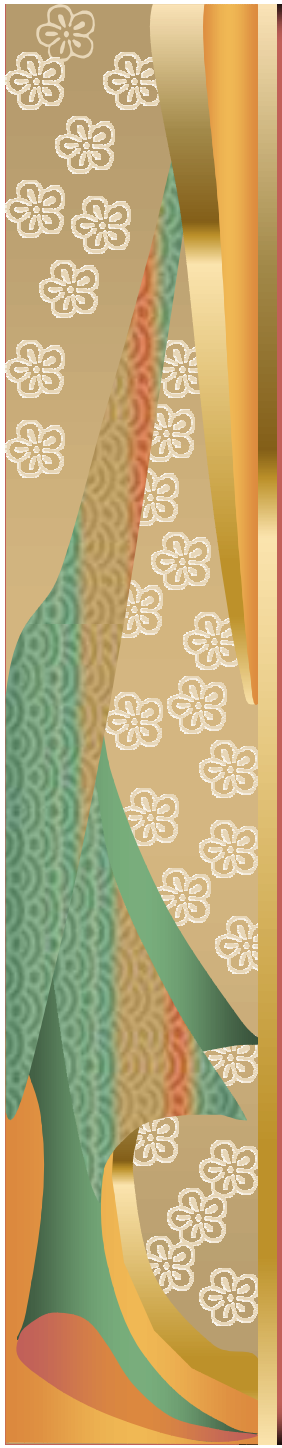


J2EE-Based Systems Development and Oracle9i JDeveloper

Bob Brown

Transentia Pty. Ltd.

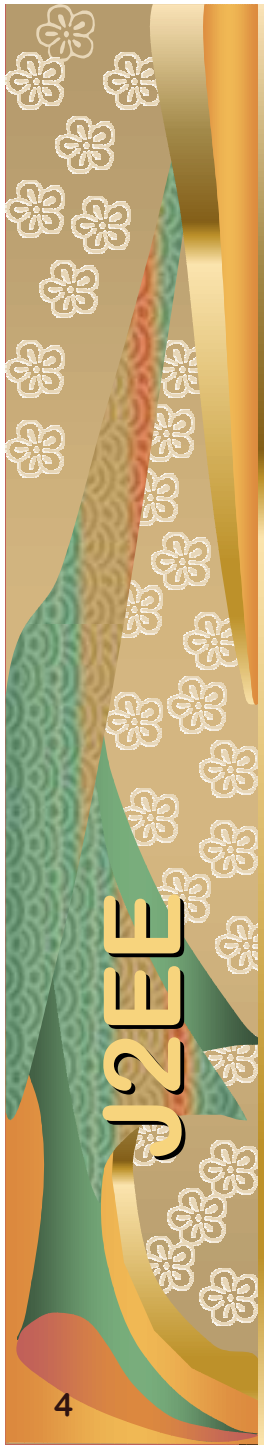
bob@transentia.com.au



Introducing the J2EE

Evolution

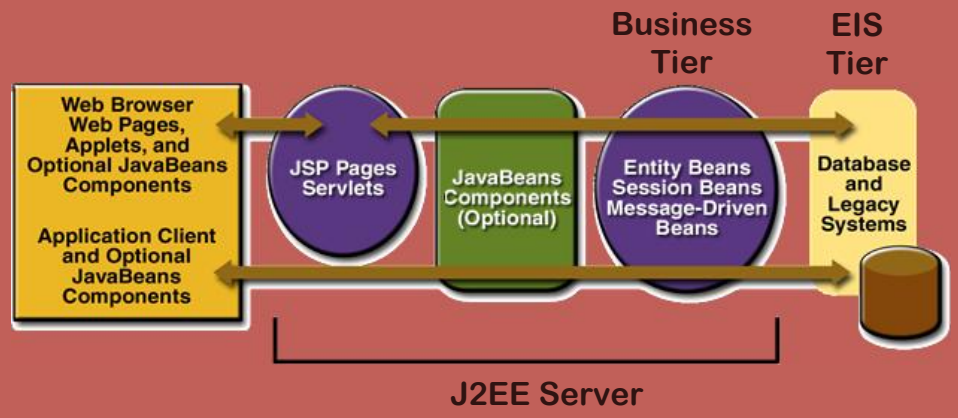
- (…Prehistory…)
- First there was HTML…
- Then CGI shell/PERL scripts
  - Date/time inserts…
  - Data-driven sites…
- Then Servlets/COM objects…
  - Replicating the last generation
    - With more efficiency
- Then JSPs/ASPs…
  - Templates and separation of concerns
- (…Still having fun with LAMP/EMACS…)
- Now EJBs/.NET components…
  - Scalable infrastructure
- (…Future…)



- Java 2, Enterprise Edition
  - Business, not technology driven
    - Internet timescales
  - Server-centric
  - Integration-oriented
  - Component-oriented
  - Simpler
    - N.B. *not* simple!
  - Tool focussed
  - Web-centric

Multitier Application Architecture

- J2EE is a framework for building *n-tier* infrastructures
  - Layered divisions of Presentation/business logic/ persistence functionality
  - J2EE focus: don't build *solutions*, build *infrastructure…*
    - *…then* build your solutions

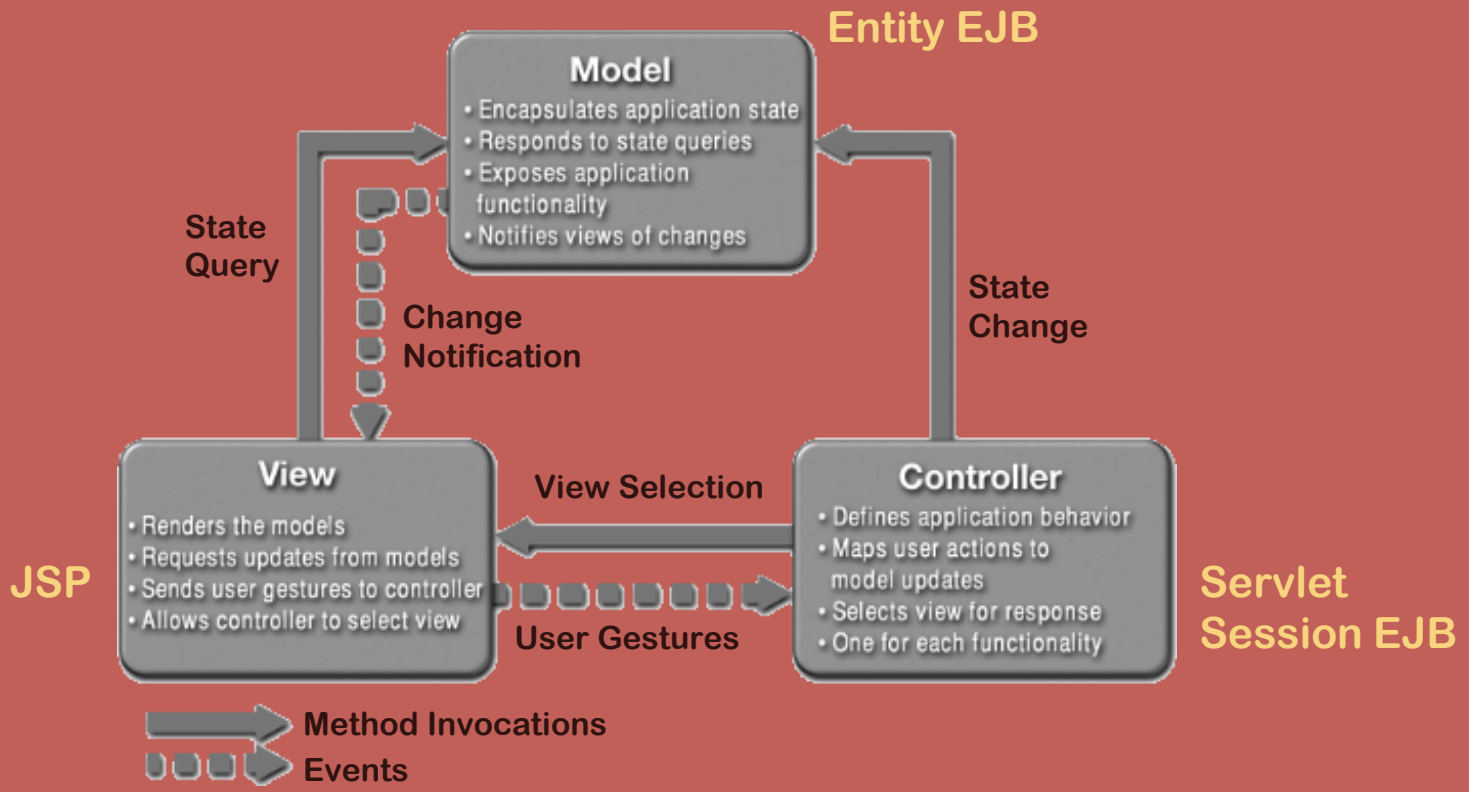


Patterns

- A design pattern describes a proven solution to a recurring design problem, placing particular emphasis on the context and forces surrounding the problem, and the consequences and impact of the solution
  - Give a common language for describing a problem and (hopefully) identifying a common solution
- Patterns underpin much of what the J2EE is about, i.e.:
  - Data Access Objects Pattern
  - Model/View/Controller (MVC) Pattern
  - Session Façade Pattern
  - Value Object Pattern
  - Page-by-Page Iterator Pattern
  - Fast-Lane Reader Pattern

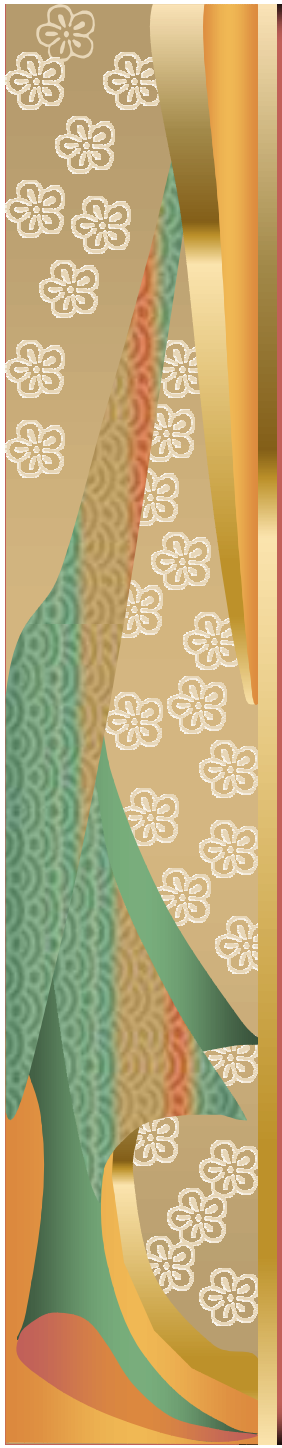
Model-View-Controller

- Design architecture underpinning J2EE
  - Sometimes called Model 2

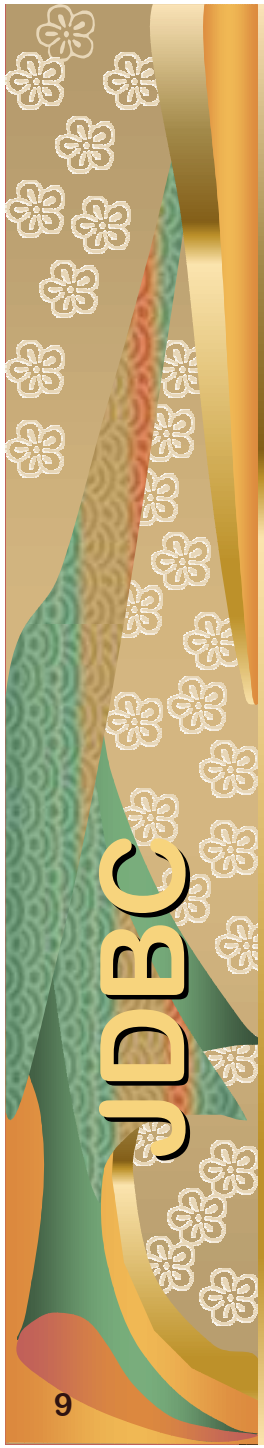


http://java.sun.com/blueprints/patterns/j2ee_patterns/model_view_controller/index.html

MVC



The J2EE Component Technologies



- Java DataBase Connectivity mechanism
- API allowing any SQL-addressable data source to be manipulated
  - Very similar to ODBC
  - JDBC architecture utilises plug-in Drivers
    - Allows four driver types
      - From native code JDBC-ODBC bridge "type 1" to 100% pure Java "type 4"
    - Vendors supply 'optimised' drivers for their products
    - A wide variety of drivers can be found on the market
      - Sun lists 30+ for Oracle alone…
  - Doesn't hide SQL
    - Still need savvy programmers
  - Driver-provided metadata allows for self-configuring applications
    - Important for an infrastructural technology

"…a core set of APIs that enables Java applications to connect to industry standard and proprietary database management systems. Using JDBC, your applications can retrieve and store information using Structured Query Language statements…"

Example: a simple SQL Select

```
import java.sql.*;

public class Query
  {
  public static void main (String[] args)
    {
    Connection conn = null;
    try
      {
      conn = DriverManager.getConnection
        ("jdbc:msql://localhost:1114/Demo","","");
      Statement stmt = conn.createStatement();
      ResultSet
        rs = stmt.executeQuery
          ("SELECT Lname FROM Customers WHERE Cnum = 8888");
      while (rs.next())
        {
        String lastName = rs.getString("Lname");
        System.out.println(lastName);
        }
      }
    catch (Exception e)
      {
      e.printStackTrace(System.err);
      }
    finally
      {
      try { conn.close();  } catch (Exception x) { /* SQUELCH! */ }
      }
    }
  }
```

JNDI

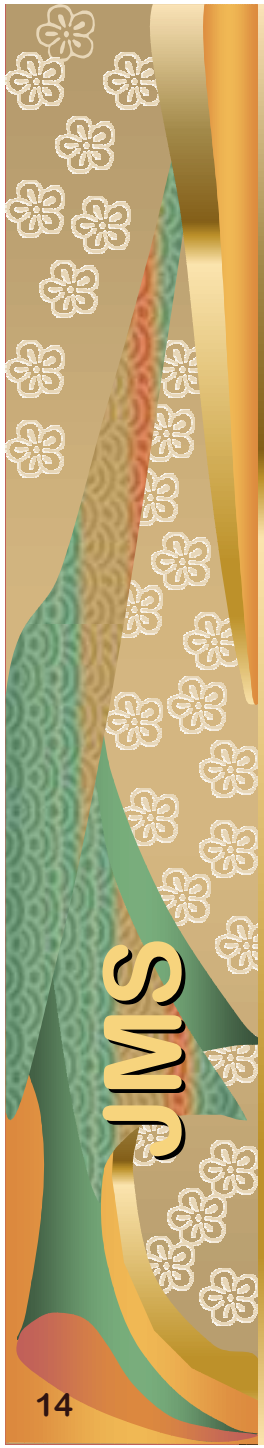
- Java Naming and Directory Interface
- Organisations typically have many directory systems
  - First stage of using something is finding it!
  - LDAP/DNS/NDS/CosNaming/Filesystem …
- API providing uniform access mechanism for disparate naming/directory services
  - Similar structure to JDBC
    - Upper-level API/lower-level driver split
- JNDI permits a graph structure
  - Subgraphs linked in from multiple domains
    - LDAP/DNS/NDS/…
- Arbitrary objects can be bound

JNDI...

```
import javax.naming.*;
import java.util.*;

public class AccountsList
  {
  public static void main(String [] args)
    {
    Hashtable env = new Hashtable();
    env.put(Context.INITIAL_CONTEXT_FACTORY,
            "weblogic.jndi.WLInitialContextFactory");
    env.put(Context.PROVIDER_URL, "t3://localhost:7001");
    try
      {
      Context ic = new InitialContext(env);
      NamingEnumeration enum = ic.listBindings("");
      while (enum.hasMore())
        {
        Binding binding = (Binding) enum.next();
        Object obj = (Object) binding.getObject();
        if (obj instanceof Account)
          {
          Account a = (Account) obj;
          System.out.print("The balance of " + a.getName() + "'s ");
          System.out.println(a.getAcc_type() + " is: " + a.getBalance());
          }
        }
      }
    catch (NamingException e)
      {
      e.printStackTrace(System.err);
      }
    }
  }
```

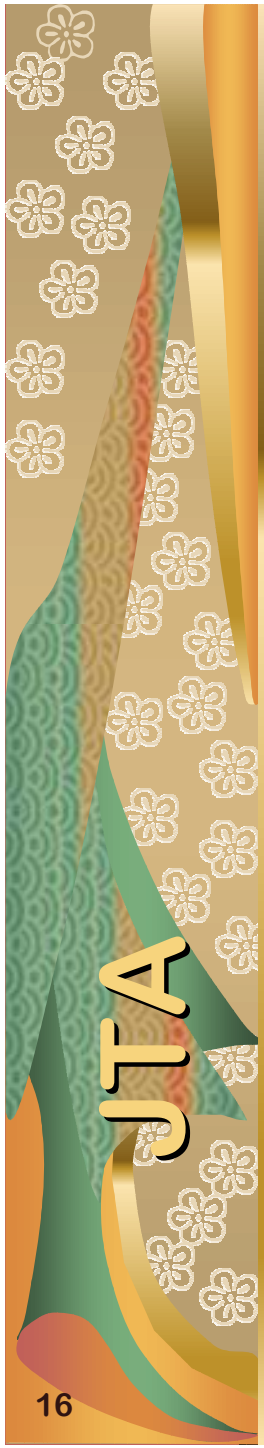
JAAS

- Java Authentication and Authorization Service
  - Available as option in J2SE 1.3, standard with J2SE 1.4
    - Thus, incorporated into the J2EE
- JAAS can be used for two purposes:
  - For authentication: determine who is currently executing Java code, regardless of whether the code is running as an application, an applet, a bean, or a servlet
  - For authorization of users to ensure that the user has the permissions required to do the desired actions
- Effectively a Java version of the standard Pluggable Authentication Module (PAM) framework
  - Supports various modules
    - JNDI
    - Unix PAM implementations
    - Windows NT
    - Kerberos
    - Keystore



- Java Messaging Service
- An API for Message Oriented Middleware
- 30+ year old technology
  - Mature, widespread
    - MSMQ/IBM MQSeries, etc.
- Important features
  - Synchronous/asynchronous
  - Connected/disconnected
  - Point-Point/Publish-Subscribe
- Capable but fiddly for developers

```
import javax.jms.*;
import javax.naming.*;

public classSimpleQueueSender
  {
  public static void main (String[] args) throws Exception
    {
    Context jndiContext = new InitialContext ();
    QueueConnectionFactory
      qcf = (QueueConnectionFactory)
        jndiContext.lookup ("QueueConnectionFactory");
    Queue queue = (Queue) jndiContext.lookup (args[0]);
    QueueConnection qconn = null;
    try
      {
      qconn= qcf.createQueueConnection ();
      QueueSessionqueueSession=
        qconn.createQueueSession (false, Session.AUTO_ACKNOWLEDGE);
      QueueSender queueSender=queueSession.createSender (queue);
      TextMessage message =queueSession.createTextMessage ();
      for (int i = 0; i < 3; i++)
        {
        message.setText ("This is message " + (i + 1));
        System.out.println ("Sending message: " + message.getText ());
        queueSender.send (message);
        }
      queueSender.send (queueSession.createMessage ());
      }
    finally
      {
      try { qconn.close (); } catch(JMSException e) {}
      }
    }
  }
```



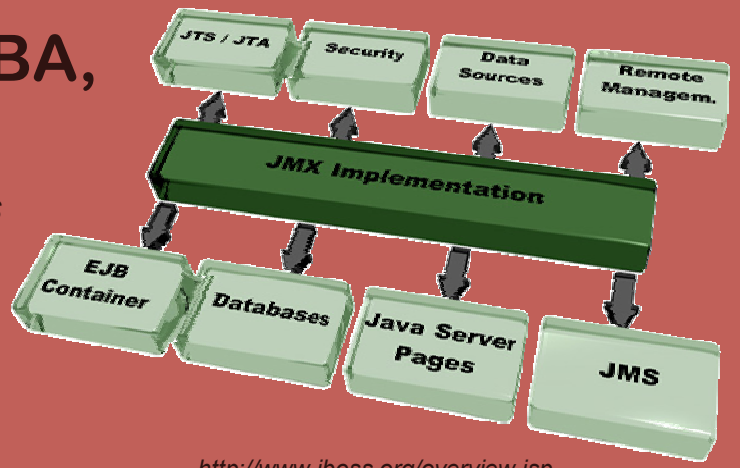
- Java Transaction API
- Developed by an industry consortium of interested players in the transaction processing/database system arenas
- Is a high level, implementation independent, protocol independent API for transactions handling
  - Allows for declarative transaction specification as well as programmatic control
  - Actual facilities available to an application depend on the underlying implementation
    - 2PC support, etc…

JMX

Java Management Extensions

- Framework for application/system management
  - An optional package addition to the J2SE
  - Widely supported
    - WebLogic, JBoss, JRun, HPAS, …
- Work in progress via the JCP to ensure JMX will integrate with existing systems
  - SNMP, TMN, CORBA, CIM/WBEM…

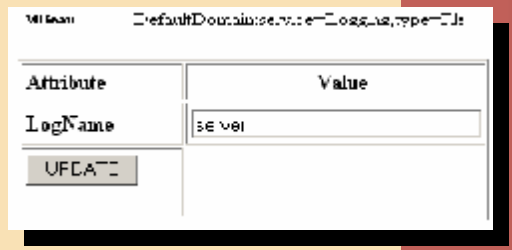
"…by design, this new standard is suitable for adapting legacy systems, implementing new management and monitoring solutions and plugging into those of the future."



http://www.jboss.org/overview.jsp

Example use: manage server logfile JSP

```
<%@ page import="javax.management.*,java.util.*" %>
<%
    try {
        ArrayList servers = MBeanServerFactory.findMBeanServer(null);
        if (servers == null)
          throw new Exception("No MBeanServer found.");
        MBeanServer server = (MBeanServer)servers.get(0);
        ObjectName objName = new ObjectName("DefaultDomain:service=Logging,type=File");
        String newvalue = (String)request.getParameter("LogName");
        if (newvalue != null && newvalue.length() > 0) {
            Attribute attr = new Attribute("LogName", newvalue);
            server.setAttribute(objName, attr );
        }
        String value = (String)server.getAttribute(objName, "LogName");
%>
MBean <%= objName.getCanonicalName() %>
<form method="post" action="jmx.jsp">
  <table align="left" border="1" width="40%" cellpadding="3">
    <tr>
      <th width="23%"> Attribute </th><th width="35%"> Value </th>
    </tr>
    <tr>
      <td><b>LogName</b></a></td>
      <td><input type="text" name="LogName" value="<%= value %>" SIZE="34%"></td>
    </tr>
    <tr>
      <td align=left><input type="submit" value="UPDATE"></td>
    </tr>
  </table>
</form>
<%
       } catch (Exception e) { out.println(e.getMessage()); }
%>
```



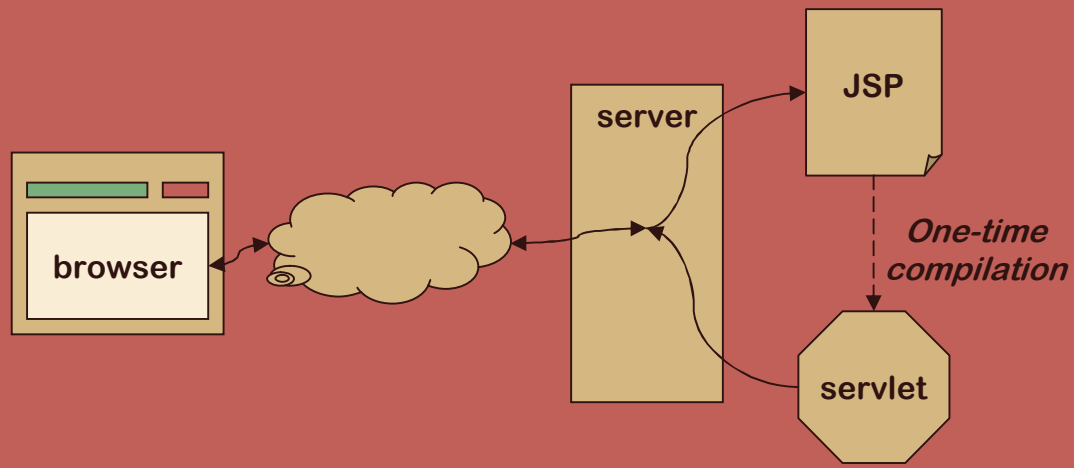
Servlets

- A standard API/mechanism for executing Java within a web server
  - A Servlet is to a server what an Applet is to a browser
- Alternative to technologies such as CGI
  - More efficient than CGI: Servlets are loaded once only (not once per request) and execute within the JVM of the web server
- Deployed against a URL
- Convenient mechanism for processing different types of HTTP requests, recovering HTML Form fields, managing state, etc.
- J2EE incorporates Servlet 2.3 which also provides support for packaging and deployment and filters

Non-persistent request counter...

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

public class CounterServlet extends HttpServlet {
  private int numReq;

  public void init(ServletConfig config)
    throws ServletException
    {
    super.init(config);
    numReq = 0;
    }

  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
  throws ServletException, IOException
    {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Sample Servlet</title></head>");
    out.println("<body>");
    out.println("<h1>Hello World!</h1>");
    out.println("This Servlet has been requested " + numReq++ + " times.");
    out.println("</body></html>");
    out.close();
    }
  }
```

- A templating mechanism for dynamic content
  - Designed with an eye on tool support
  - Presentation material (html, xml, wml) plus embedded dynamic actions
    - An “inside-out” Servlet
    - Translated to a Servlet
      - Typically as a response to *first* incoming request



Separate presentation / processing

```
<html>
<head>
<title>Form Handling JSP</title>
</head>
<body bgcolor="#ffffcc">
<% if (request.getParameter("name")==null &&
       request.getParameter("email")==null) { %>
<center>
<h2>User Info Request Form</h2>
<form method="POST" action="process.jsp">
  Your name: <input type="text" name="name" size=26><br />
  Your email: <input type="text" name="email" size=26><br />
  <input type="submit" value="Process">
</form>
</center>
<% } else { %>
<%! String name, email; %>
<%
name = request.getParameter("name");
email = request.getParameter("email");
%>
<p>
You have provided the following info:<br />
<b>Name</b>: <%= name %><br>
<b>Email</b>: <%= email %>
</p>
<% } %>
</body>
</html>
```

```
<html>
<head>
<title>useBean action test page</title>
</head>
<body>
<h1>useBean action testing page</h1>
<form method="post" action="usebeanjsp.jsp">
  <p>What is your name ?
  <input type="text" name="name">
  </p>
  <p>What is your age ?
  <select name="age">
    <option value="under 20">under 20
    <option value="20 to 30">20 to 30
    <option value="over 30">over 30
  </select></p>
  <input type="submit">
</form>
</body>
</html>
```

```
package jspdemo;

public class AgeBean {
  private String name;
  private String age;
  public void setName(String name)
    { this.name = name; }
  public String getName()
    { return name; }
  public void setAge(String age)
    { this.age = age; }
  public String getAge()
    { return age;  }
  }
```

```
<jsp:useBean id="ageBean" scope="page" class="jspdemo.AgeBean">
  <jsp:setProperty name="ageBean" property= "*" />
</jsp:useBean>
<html>
<head>
  <title>useBean action test result</title>
</head>
<body>
  <h1>useBean action test result</h1>
  <p>Hi, <jsp:getProperty name="ageBean" property="name"/>.</p>
  <p>Your age bracket is: </p>
  <p><jsp:getProperty name="ageBean" property="age" /></p>
</body>
</html>
```

TagLibs

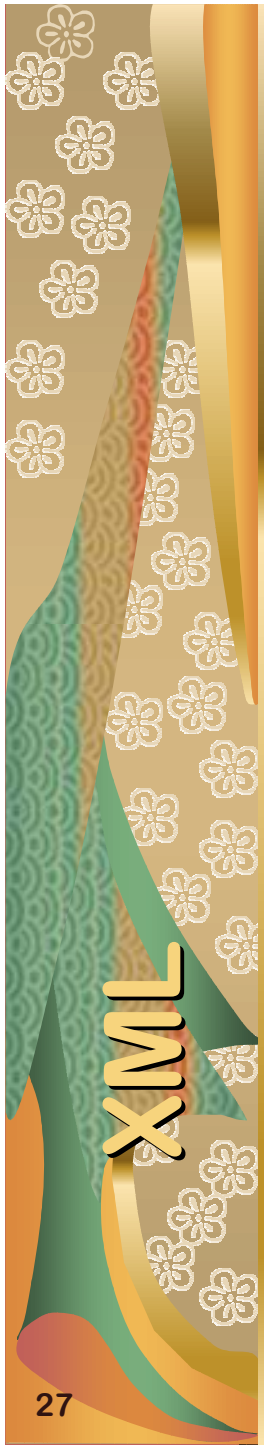
- Developer can define custom tags and provide an implementation for the tags' activities
  - Written in Java, deployed using XML
- Allows the better separation of presentation and processing
  - Allow for even better tool support
  - Permits clear division of job roles
- Develop infrastructure, not just solutions
  - Supports the development of an aftermarket for tags
    - JCP STL, apache taglibs, etc.

TagLib...

```
<%@ taglib uri="counter" prefix="counter" %>
<html>
<head><title>JSP Tag Library Counter Example</title></head>
<body>
  <h2>JSP Counter</h2>
  <counter:increment/>
  <p> This page has been visited <counter:display/> times!
  <p> <a href="./pagehits.jsp">Click Here</a> to visit again!
</body>
</html>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib
        PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.1//EN"
          "http://java.sun.com/j2ee/dtds/web-jsptaglibrary_1_1.dtd">

<taglib>
  <tlibversion>1.0</tlibversion>
  <jspversion>1.1</jspversion>
  <shortname>counter</shortname>
  <info>
    This taglib provides tags to increment a file-based hit counter
    and display the current count.
  </info>
  <tag>
    <name>display</name>
    <tagclass>examples.jsp.tagext.counter.Display</tagclass>
    <bodycontent>empty</bodycontent>
  </tag>
  <tag>
    <name>increment</name>
    <tagclass>examples.jsp.tagext.counter.Increment</tagclass>
    <bodycontent>empty</bodycontent>
  </tag>
</taglib>
```

```
package examples.jsp.tagext.counter;
import java.io.*;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.http.*;

public class Display extends TagSupport {
  private static String IMAGE_DIR="images/numbers/";

  public int doStartTag() throws JspException {
    File tempDir = (java.io.File) pageContext.getServletContext().
       getAttribute("javax.servlet.context.tempdir");
    File countFile = new File(tempDir, "count.tmp");
    String countStr = String.valueOf(Count.getCount(countFile));
    try {
      JspWriter out = pageContext.getOut();
      int i = 0;
      while(i < countStr.length()) {
        out.print("<img src=\""+IMAGE_DIR+countStr.charAt(i)+
                  ".gif\" height=40 >");
        i++;
      }
    } catch(IOException ioe) {
      throw new JspException("Failed to insert counter display");
    }
    return(SKIP_BODY);
  }
}
```



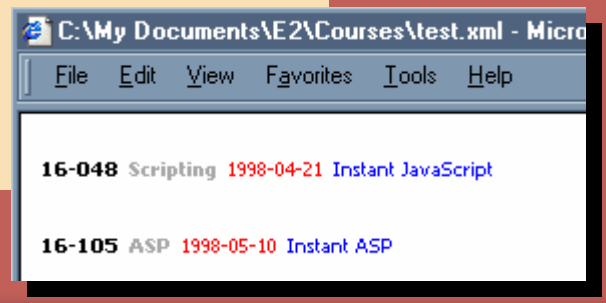
eXtensible Markup Language

- A language for structuring data
  - Supports the concept of a ‘self-describing’ document
    - Essential for informal e-commerce interactions
    - Excels where data has to be interchanged across heterogeneous boundaries
- Also a plethora of allied technologies
  - Schema, XSL, XLink, …
- J2EE has/will incorporate XML into its core via an alphabet soup of technologies:
  - Java API for XML Messaging (JAXM)
  - Java API for XML Processing (JAXP)
  - Java API for XML Registries (JAXR)
  - Java API for XML-based RPC (JAX-RPC)
  - SOAP with Attachments API for Java (SAAJ)
  - Java Architecture for XML Binding (JAXB)
- Forms the foundation of WebServices
  - A key focus of J2EE 1.4 (forthcoming)

Example: XML and XSLT

```
<?xml version = "1.0" ?>
<xsl:stylesheet
   xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <html>
      <body>
      <xsl:for-each select="BOOKLIST/ITEM">
        <p><font face="Tahoma,Arial,sans-serif">
        <xsl:apply-templates/>
        </font></p>
      </xsl:for-each>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="CODE">
    <b><font size="0"><xsl:value-of /></font></b>
  </xsl:template>
  <xsl:template match="CATEGORY">
    <b><font size="1" color="darkgray">
    <xsl:value-of /></font></b>
  </xsl:template>
  <xsl:template match="RELEASE">
    <font size="0" color="red"><xsl:value-of /></font>
  </xsl:template>
  <xsl:template match="TITLE">
    <font size="1" color="blue"><xsl:value-of /></font>
  </xsl:template>
  <xsl:template match="SALES">
  </xsl:template>
</xsl:stylesheet>
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
                 href="booklist.xsl"?>
<BOOKLIST>
 <ITEM>
  <CODE>16-048</CODE>
  <CATEGORY>Scripting</CATEGORY>
  <RELEASE>1998-04-21</RELEASE>
  <TITLE>Instant JavaScript</TITLE>
  <SALES>375298</SALES>
 </ITEM>
 <ITEM>
  <CODE>16-105</CODE>
  <CATEGORY>ASP</CATEGORY>
  <RELEASE>1998-05-10</RELEASE>
  <TITLE>Instant ASP</TITLE>
  <SALES>297311</SALES>
 </ITEM>
</BOOKLIST>
```



XML...

Java Connector Architecture

- A standard architecture for integrating Java applications with existing back-end Enterprise Information systems
  - Allows EIS systems to be reached at a “high level” from the J2EE platform
    - Akin to using JDBC to access ‘raw’ data
- More extensive than JDBC, covers:
  - Resource adapters
  - Data mapping
  - Messaging brokers
  - Workflow
- Good industry support
  - Early days, however…

JCA

Example: employee count from an EIS using the *Common Client Interface* API

```
...
int count = -1;
try {
    ConnectionSpec spec = new CciConnectionSpec(user, password);
    Connection con = cf.getConnection(spec);

    Interaction ix = con.createInteraction();
    CciInteractionSpec iSpec = new CciInteractionSpec();
    iSpec.setSchema(user);
    iSpec.setFunctionName("EMPLOYEECOUNT");
    RecordFactory rf = cf.getRecordFactory();
    IndexedRecord iRec = rf.createIndexedRecord("InputRecord");

    Record rec = ix.execute(iSpec, iRec);
    Iterator iter = ((IndexedRecord)rec).iterator();
    while(iter.hasNext()) {
        Object obj = iter.next();
        if(obj instanceof Integer) {
            count = ((Integer)obj).intValue();
        }
    }
    con.close();
}
catch(Exception e) {
    e.printStackTrace();
}
System.out.println("Employee count is: " + count);
...
```

- Enterprise JavaBeans
  - Component architecture for the enterprise supporting large-grained component re-use
    - Allowing off-the-shelf components to be written
    - Configured during deployment
    - Run on any platform
  - Tool oriented
  - EJBs live within a container that provides complete lifecycle support
    - Packages security, transactions, persistence, relationships, reliability, etc.
    - “Control by interposition”
  - Should not be confused with ‘plain’ JavaBeans

“The Enterprise JavaBeans architecture is a component architecture for the development and deployment of object-oriented distributed enterprise-level applications. Applications written using the EJB architecture are scalable, transactional and multi-user secure.”

EJBs...

- Three types of EJB:
  - Session Bean
    - Typically represents transient process-oriented activities
    - Two subtypes
      - Stateful
      - Stateless
  - Entity Bean
    - Represents persistent real-world entities
    - Two major persistence flavours
      - Container managed
        - » May include relationship management
      - Bean Managed
  - Message-Driven Bean
    - Packages the JMS into the J2EE world
    - Architecturally similar to Stateless Session Bean
      - Invoked asynchronously
      - Never interacts with client directly
        - » Always via Topic or Queue

- Comprised of 4+ files
  - Bean Class
  - Various Interfaces
  - XML Deployment Descriptor(s)

```
// file: TraderHome.java
import java.rmi.RemoteException;
import javax.ejb.*;

public interface TraderHome extends EJBHome {
  Trader create() throws CreateException, RemoteException;
}
```

```
// file: Trader.java
import java.rmi.RemoteException;
import javax.ejb.EJBObject;

public interface Trader extends EJBObject {
  boolean buy (String stockSymbol, int shares)
    throws RemoteException;
  boolean sell (String stockSymbol, int shares)
    throws RemoteException;
}
```

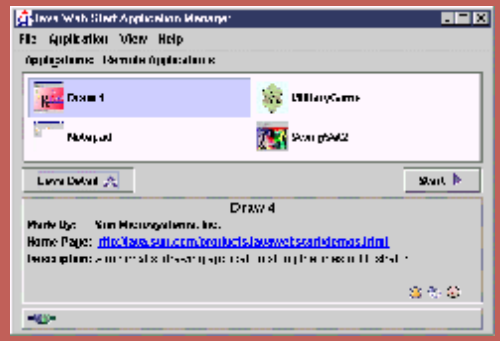
EJBs...

```
// file: TraderBean.java
import javax.ejb.*;
import javax.naming.*;

public class TraderBean implements SessionBean {
  private int tradeLim;

  public void ejbActivate() { }
  public void ejbPassivate() { }
  public void ejbRemove() { }
  public void setSessionContext(SessionContext ctx) { }

  public void ejbCreate () throws CreateException {
    try {
      InitialContext ic = new InitialContext();
      tradeLim = ((Integer) ic.lookup("java:comp/env/tradeLim")).intValue();
    }
    catch (NamingException ne) {
      throw new CreateException("Failed to find environment value. "+ne);
    }
  }

  public boolean buy(String stockSymbol, int shares) {
    if (shares > tradeLimit)
      return false;

    shares = tradeLimit;
    return true;
    }

  public boolean sell(String stockSymbol, int shares)
    { /* … elided … */ }
}
```

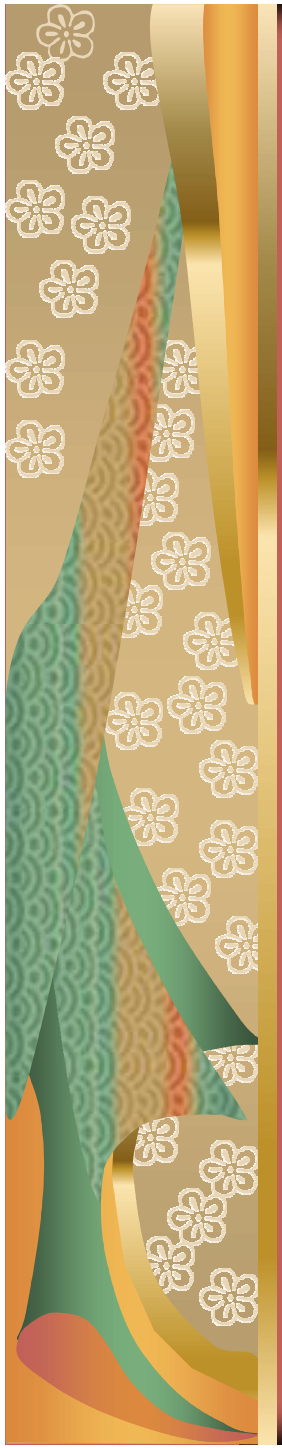
```
<!- file: ejb-jar.xml ->
<?xml version="1.0"?>
<!DOCTYPE ejb-jar PUBLIC
          '-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans 2.0//EN'
          'http://java.sun.com/dtd/ejb-jar_2_0.dtd'>

<ejb-jar>
  <enterprise-beans>
    <session>
      <ejb-name>statelessSession</ejb-name>
      <home>TraderHome</home>
      <remote>Trader</remote>
      <ejb-class>TraderBean</ejb-class>
      <session-type>Stateless</session-type>
      <transaction-type>Container</transaction-type>
      <env-entry>
        <env-entry-name>tradeLim</env-entry-name>
        <env-entry-type>java.lang.Integer</env-entry-type>
        <env-entry-value>500</env-entry-value>
      </env-entry>
    </session>
  </enterprise-beans>
  <assembly-descriptor>
    <container-transaction>
      <method>
        <ejb-name>statelessSession</ejb-name>
          <method-name>*</method-name>
      </method>
      <trans-attribute>Supports</trans-attribute>
    </container-transaction>
  </assembly-descriptor>
</ejb-jar>
```

Java Web Start

- A technology for simplifying deployment of Java *applications* to a desktop
  - As Jar files
- Allows a user to launch and manage applications right off the Web
  - Similar to applets but with greater functionality
- Simplifies version checking/upgrades
- Security aware
- Configured via manager tool/XML



JWS



J2EE Development with Oracle9i JDeveloper

JDeveloper 9i

- Oracle's flagship IDE
  - Free download
- Complete re-write from JDeveloper 3
  - No longer a re-badged JBuilder
- Introduced May 2002
- Extensive feature set
- Quite standards-oriented
  - But with lots of Oracle goodies to make life easier
  - Will be revised regularly as standards evolve

JDeveloper 9i...

- 100% pure Java
  - Supports Unix/Linux, windows
- Oracle goodies
  - Integrated SQL/PL/SQL support
  - Business Intelligence Beans
  - UIX framework for Oracle Browser look&feel
  - Business Components for Java
    - Pattern framework
- Many wizards
- Partial UML modelling support
  - Only 2 diagram types
    - Currently
  - Tailored to Oracle's BC4J
- Support for J2EE
  - Oracle9iAS Containers for J2EE (OC4J)
  - Not completely up-to-the-latest specifications, though
    - E.g. only EJB1.1
- XML
  - Schemas, XSL, XHTML
  - WebServices
- Extensive 2-tier development support
- Integrated debugging
- Integrated profilers and CodeCoach
- Support for WebDAV and source-code management
- Some support for WebLogic Server 6.x
  - A bit buggy...

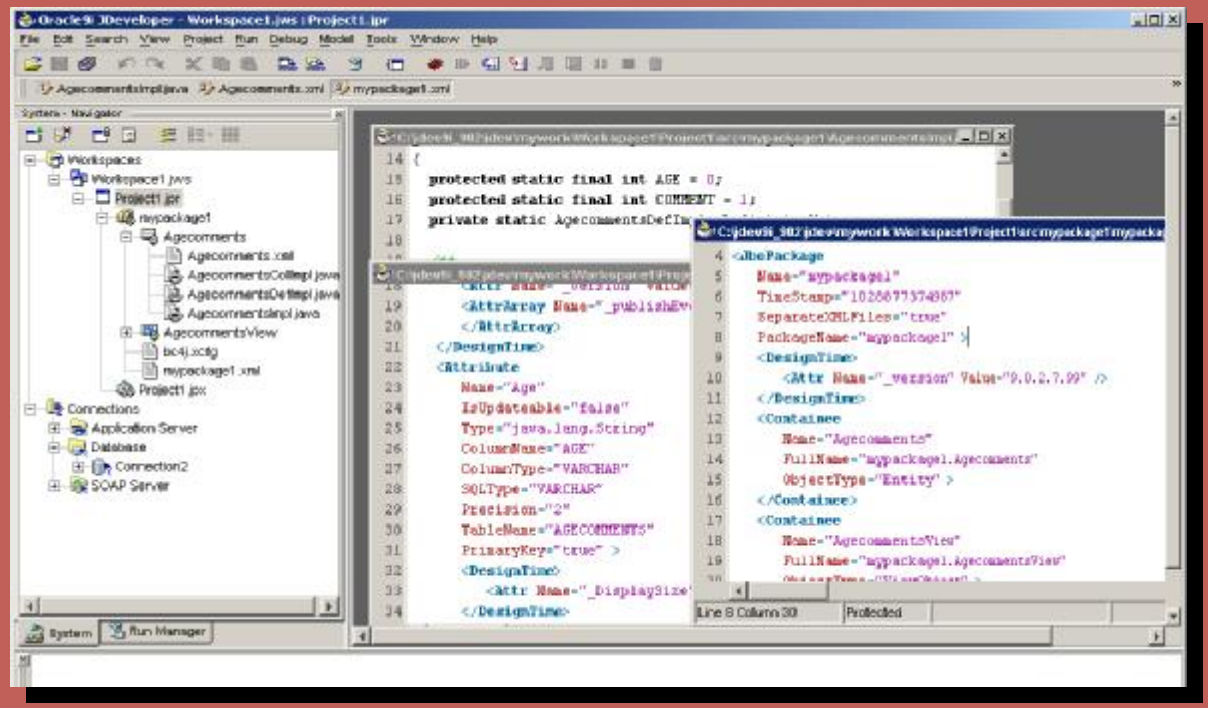
BC4J Components

- Based on Java Enterprise Blueprints
  - Plus various additions
- JDO approach
  - Can map to EJBs/CORBA/…
- Higher-level collection of patterns expressed through wizards to make life easier
  - Query Bean
    - Implements (among other things) the Data Access Objects, Fast-Lane Reader, and Page-by-Page Iterator patterns
  - Generic Value Object Bean
    - Implements the Value Object pattern
  - MVC Application Object Bean
    - Implements the data model in a Model/View/Controller pattern
  - Generic JSPData Binding Tag Library

BC4J Components…

Declarative approach

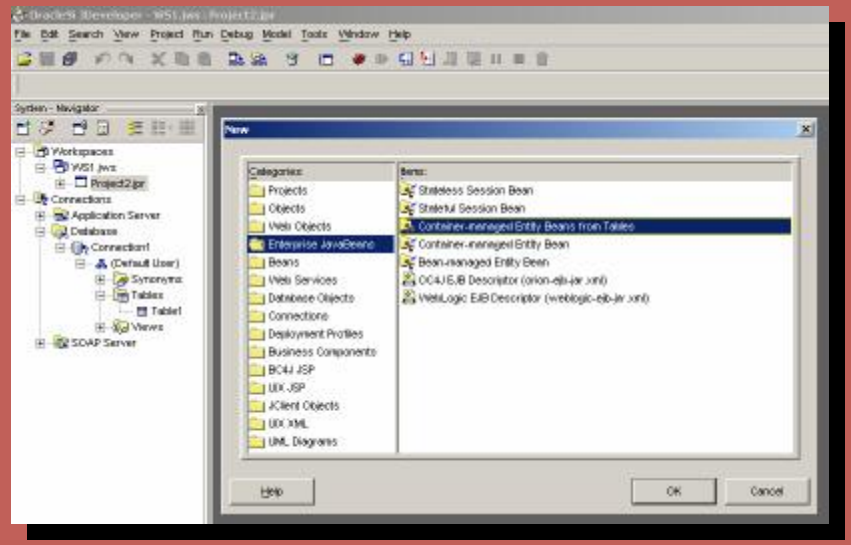
XML deployment /configuration



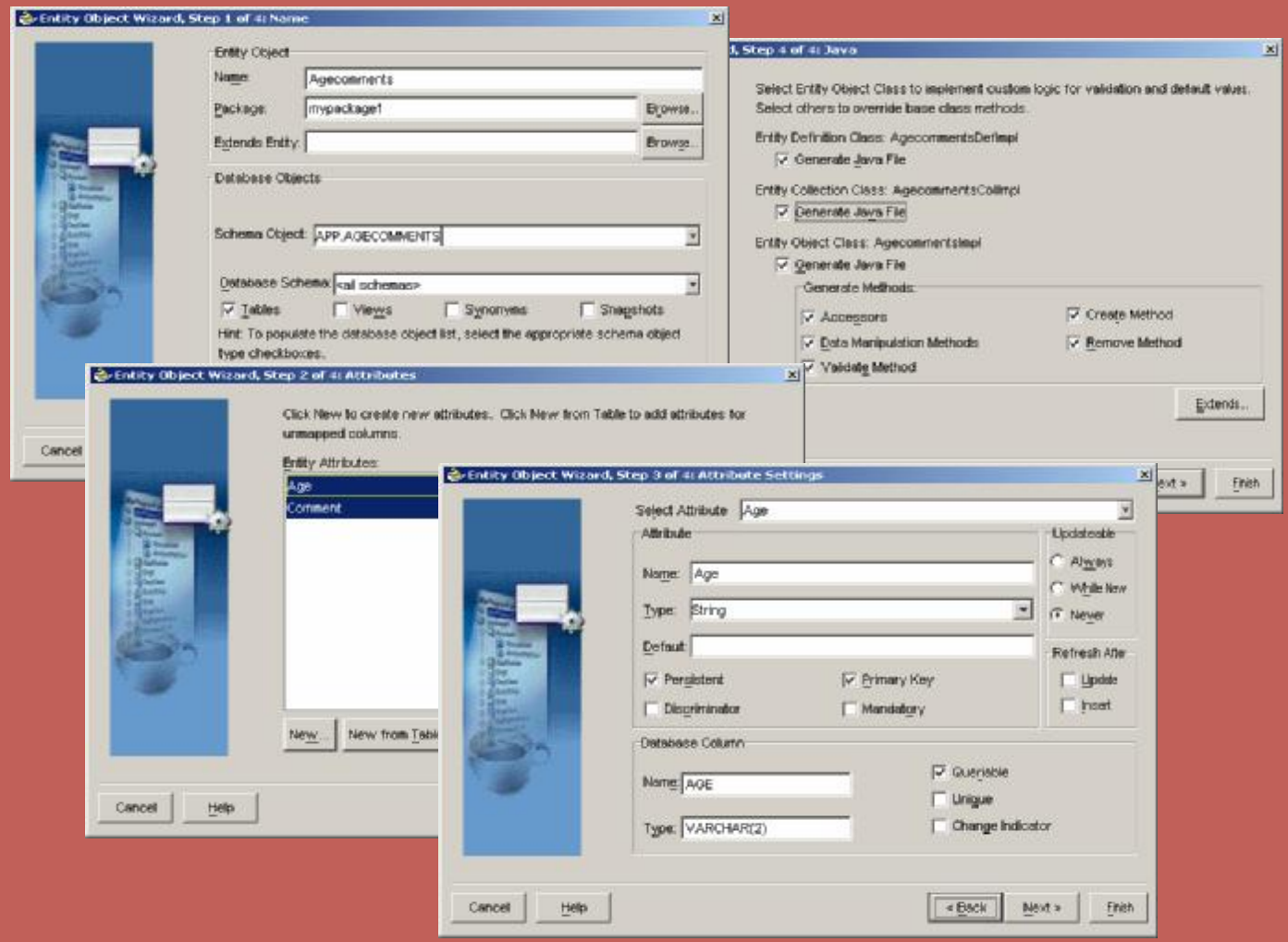
InfoWorld rates BC4J as having no ‘cons’

“Using BC4J to create reusable business components can reduce development time, allowing you to bring applications to market faster. You can also introduce efficiencies to your programming team by creating libraries of objects once other developers can repurpose those objects in similar applications.”

- Entity EJBs
  - Represents a business object in an underlying persistent storage mechanism
  - Performs an Object-Relational mapping
    - Simply: proxies for a tuple in a table
  - Has a unique identity
    - Identified by a primary key
- Various wizards support Entity EJBs
  - Top-down development of new functionality
  - Bottom-up mapping to existing persistent data



Also wizards for BC4J Entity Objects

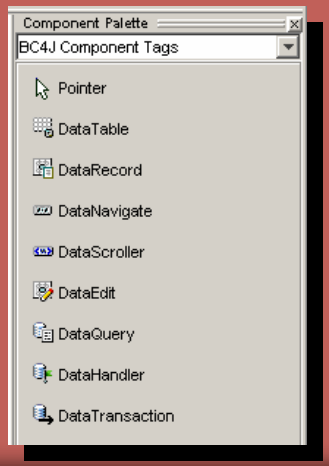


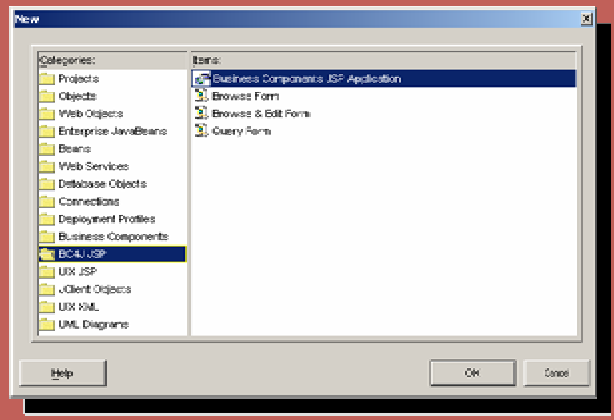
Encapsulating Business Logic

- Plain J2EE Session EJBs available
  - Simple wizard-driven creation
- BC4J also provides two different kinds of EJB Session Bean Façades:
  - A “Service Session Bean” that exploits BC4J exclusively inside its own private implementation, but does not expose these framework objects to the remote client
    - Thus fits in with a ‘vanilla’ J2EE world
  - An enhanced “AppModule Session Bean” that provides full remote access to the BC4J design pattern components
    - Thus simplifying life for the developer

JSPs and tag libraries

- Provides various wizards for 'vanilla' JSP/taglibs
- Also provides framework-oriented *BC4J JSP application*
  - Pre-canned JSP solutions
  - Template-driven approach for data-bound JSPs
    - Bind to various BC4J pre-canned components
    - Bind to AppModule

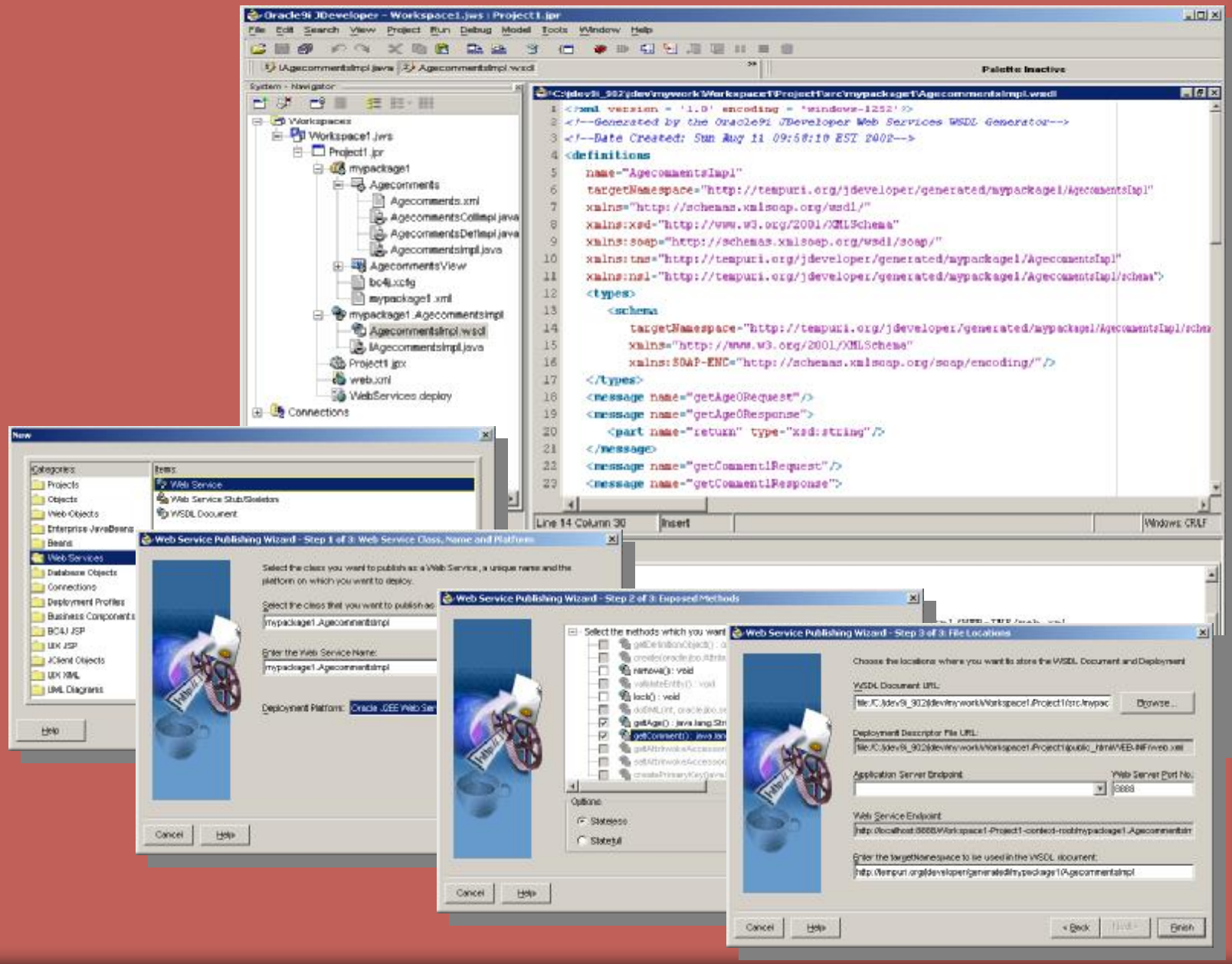




WebServices

- Being introduced into the J2EE through J2EE 1.4 and JSR 109
- JDeveloper can ‘wrap’ most components as WebServices
  - BC4J & EJBs
- Oracle provides a WebServices tag library with OC4J to make life easier
  - Can use with JDeveloper, obviously

Wizard wonderland!



Example: check an item on ebay

```
<%@ page contentType="text/html"%>
<%@ taglib uri="http://xmlns.oracle.com/j2ee/jsp/tld/ws/wstaglib.tld"
           prefix="ws" %>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; ">
</HEAD>
<BODY>
<%
 String itemID = request.getParameter("itemID");
%>
<ws:webservice id="ebay"
    wsdlUrl="http://www.xmethods.net/sd/2001/EBayWatcherService.wsdl"
    binding="eBayWatcherBinding"
    soapLocation="http://services.xmethods.net:80/soap/servlet/rpcrouter"
    scope="page">
  <ws:property name="http.proxyHost" value="www-proxy.us.oracle.com"/>
  <ws:property name="http.proxyPort" value="80"/>
</ws:webservice>
<ws:invoke id="price" webservice="ebay" operation="getCurrentPrice">
  <ws:part name="auction_id" value="<%=itemID%>"/>
</ws:invoke>
<B>Action price for eBay Item # <%=itemID%> is :</B>
<P>
$<%= price%>
@
<%= new java.util.Date()%>
</P>
</BODY>
</HTML>
```

Whether to "buy in" to all these BC4J/Wizards/goodies…

"ay, there's the rub…"

"There are a few areas where JDeveloper can lock unsuspecting developers into an Oracle-only solution (the Web Services integration frameworks and the Business Objects support both require Oracle provided libraries), but there's nothing inherently wrong with a server vendor making it easy to use their servers."

"Although you may not be able to create complex production applications using only the wizards, you will be able to build working prototypes which can be expanded."

"Experienced Java Developers will tend to avoid the wizards and to build their code by hand…Code should be generated from the wizards where possible…However, if you ignore the wizards and write your code as you please…It is easy to write code that negates much of the power of JDeveloper."

"I'd like to see Oracle submit a BC4J JSR or utilize another mechanism to guarantee that I won't be locked into a vendor-proprietary solution if I use the BC4J framework."

BC4J (Business Components for Java) is very hot stuff. It lets you build reusable components like blocks in Forms. It's all in JDeveloper and is very reminiscent of Designer module APIs and table APIs.

Cat Among Pigeons Time

- XDoclet
  - A generic Java tool that lets you create custom Javadoc @tags and based on those @tags generate source code or other files (such as xml deployment descriptors)
  - “Continuous Reconfiguration”
    - Don’t have to worry about outdating deployment meta-data whenever you touch the code
  - Integrated into the build process with Ant

“This free open source solution can simplify your EJB development, allowing you to work just with a bean class and have the interfaces and descriptors generated for you. This framework is growing, and isn't just for EJBs.”

XDoclet...

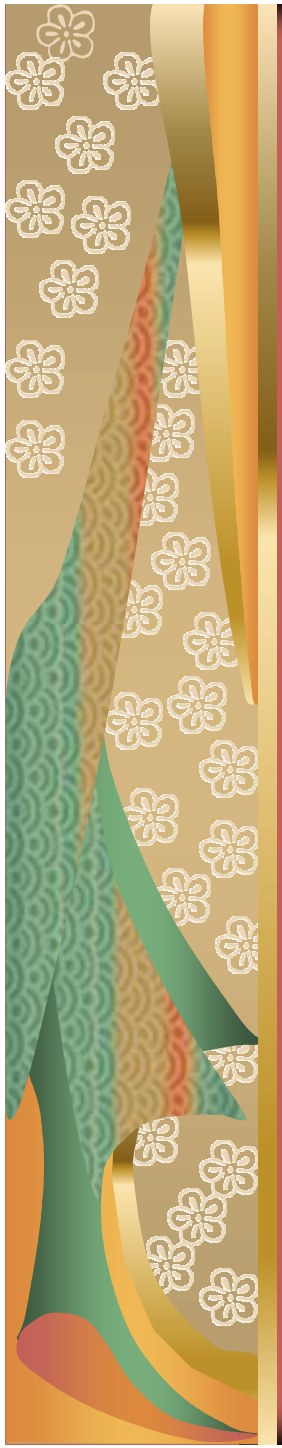
```
/**
 *   @ejb:bean type="Stateless"
 *             name="MyXDocletEJB"
 *             jndi-name="the-ejb-name"
 *             display-name="The EJB"
 *
 *   @ejb:env-entry name="parameter" type="java.lang.String" value="some.data"
 */
public class EJB implements SessionBean {


  /**
   *  @ejb:interface-method view-type="remote"
   */
  public void aMethod(String someParam) throws RemoteException {
```

```
<target name="ejbdoclet" depends="prepare">

  <taskdef name="ejbdoclet"
           classname="xdoclet.ejb.EjbDocletTask"
           classpath="${java.class.path};${xdoclet.jar.path};
           ${log4j.jar.path};${ant.jar.path}"/>

  <ejbdoclet sourcepath="${java.dir}"
             destdir="${generated.java.dir}"
             ejbspec="2.0">

    <fileset dir="${java.dir}">
      <include name="**/EJB.java" />
    </fileset>

    <remoteinterface/>
    <homeinterface/>
    <deploymentdescriptor destdir="${build.dir}/ejb/META-INF"/>
    <orion destdir="${build.dir}/ejb/META-INF" />

  </ejbdoclet>
</target>
```

The Future for JDeveloper

- Improved/Updated support for
  - JSP
  - Struts
  - EJB
    - When revised OC4J is incorporated
  - UML

"JDeveloper is Oracle's strategic development tool, and as such will continue to evolve quickly, especially in the modeling field."



Resources

Resources

- Where to find more…
  - http://java.sun.com/blueprints/enterprise/index.html
  - http://otn.oracle.com/products/jdev/
  - http://otn.oracle.com/products/jdev/htdocs/j2ee_bc4j.html
  - http://www.dulcian.com/papers
    - *Don't be a Muggle! Be a JDeveloper 9i Wizard!*
  - http://www.fawcette.com/javapro/2002_05/magazine/departments/productreviews/default_pf.asp
  - http://www-106.ibm.com/developerworks/java/library/j-webstart/?dwzone=java
  - http://xdoclet.sourceforge.net