

---

# Technology Analysis Profile: WebSphere Studio Version 5.0

---

*This document is a discussion paper that brings into focus our experiences and observations of WebSphere Studio Application Developer Version 5.0.*

***for further information contact:***

***Loren Abdulezer  
Evolving Technologies Corporation  
loren@evolvingtech.com***

---

## 1 Introduction & Background

---

*This document briefly outlines some of our experiences in working with the WebSphere Studio Application Developer Version 5.0 product recently released by IBM.*

It could be said that the IT community has been pursuing the goal of true enterprise computing where the many diverse applications, whether legacy based, can seamlessly work together. Some have hailed Web Services as the answer. While the jury is still out on whether Web Services is the solution for end-to-end enterprise computing, it seems clear that the industry is headed in the right direction.

### **Frameworks for Enterprise Development**

Well designed and well managed enterprise computing applications, whether or not Web Services based, need a development framework that supports the complex requirements of enterprise applications.

Enterprise computing applications are normally born in large development shops with teams of developers and staff from all walks including technical writers, network engineers, security analysts, quality control testers, web architects, and so forth. Such teams are concerned as much about process as they are about product. The complexity of enterprise applications are enormous. There is a very real need to capture and retain the working knowledge and techniques amassed during the application development process. There is a need to have a strong handle on the life cycle management process.

---

## Some Note Worthy Features

---

From company to company, the needs of these application development shops are varied. Some are forced to contend with the marriage of immiscible technologies which were never designed to work together. Others may have stringent legal and reporting requirements that can change very rapidly, in fact, more rapidly than traditional product development cycles. Others are concerned with real time processing and maintaining reliability in transaction based systems.

### End-to-End Strategies in Application Development

What is the ideal development framework for these markedly different types of applications? The answer can't be one single system. What is required is an open framework that can be moulded to the specific requirements of each development shop.

WebSphere Studio Application Developer and its related products fits into this scheme for such an open framework. It was designed, from the ground up, to incorporate third party and custom designed plug-ins, allow for customized roles and perspectives, utilize Software Configuration Management to accommodate versioning control needs, and many others.

The overall idea in mind is to have an open framework so that developers are free to incorporate any or all available Best-of-Breed products and best practices into whatever solution is required by individuals for their companies and organizations.

## 2 Some Note Worthy Features

---

WebSphere Studio Application Developer has some notable features, which are listed in the following table:

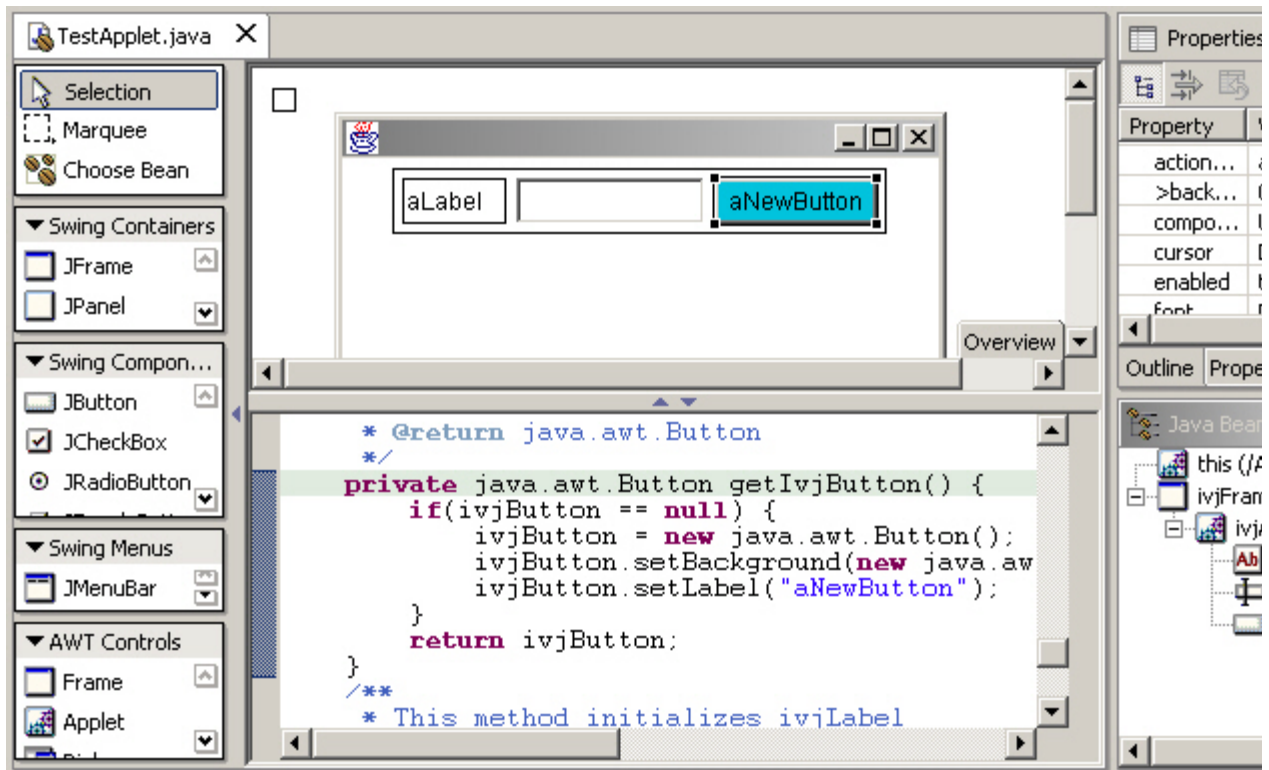
---

**TABLE 1.** Representative features of WebSphere Studio Application Developer Version 5.0

<b>WebSphere Studio Application Developer features</b>
retains and enhances the essential features VisualAge for Java including use of the Scrapbook, and incremental compilation, quick and easy code browsing, refactoring and Team Development facilities.
ability to design, develop and deploy for VMs of your choosing
has a separate class loader for each project
introduces the notion of roles and customizable perspectives
incorporates integrated and easily configurable test environment
definable macro templates
custom definable markers
custom definable plug-ins

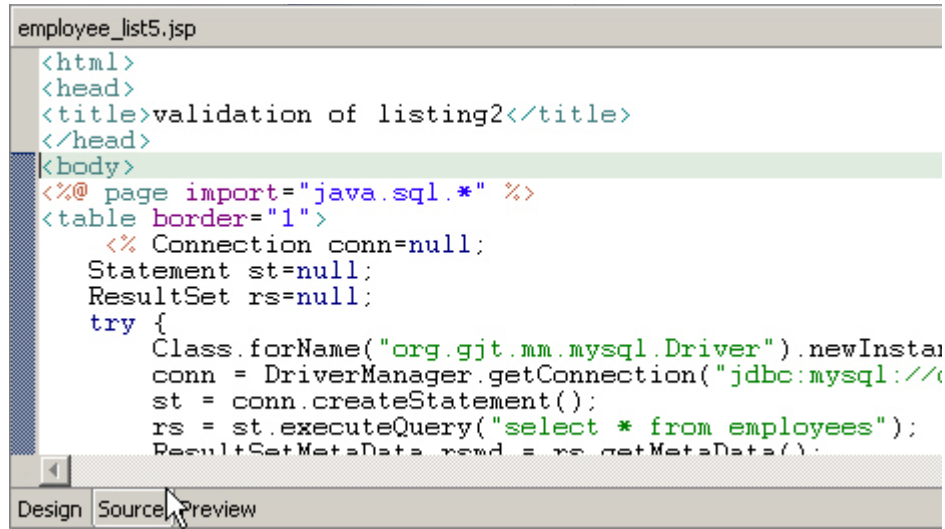
### Building a Graphical User Interface

Sometimes a Picture is worth a thousand lines (of code). Specifically, as you highlight and make changes to the graphical representation of your components, the underlying source code is automatically revised. The converse also true. As you change your underlying source code, your graphical components are instantly revised to reflect the source code changes.



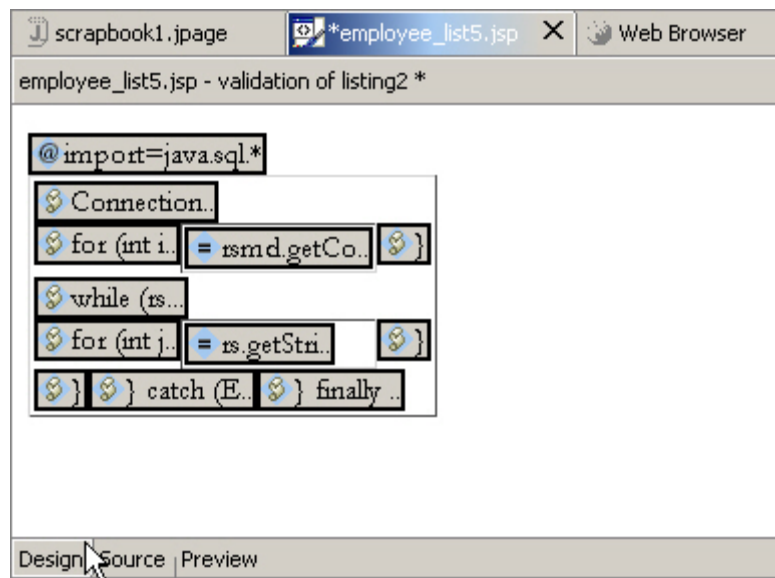
### Web Applications - A Sample JSP Application

Imagine you have a three tier application which presents information to the end user via a JSP presentation layer. On the back end the application server gets the content from a relational database. For the purposes of illustration, we're not going to do anything complex here. Because the query is simple, we don't need to bring in XML. We can make do with a simple SQL query. You can compose and view source code in a seemingly conventional looking text editor.



```
employee_list5.jsp
<html>
<head>
<title>validation of listing2</title>
</head>
<body>
<%@ page import="java.sql.*" %>
<table border="1">
  <% Connection conn=null;
  Statement st=null;
  ResultSet rs=null;
  try {
    Class.forName("org.gjt.mm.mysql.Driver").newInstance();
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/");
    st = conn.createStatement();
    rs = st.executeQuery("select * from employees");
    ResultSetMetaData rsmd = rs.getMetaData();
```

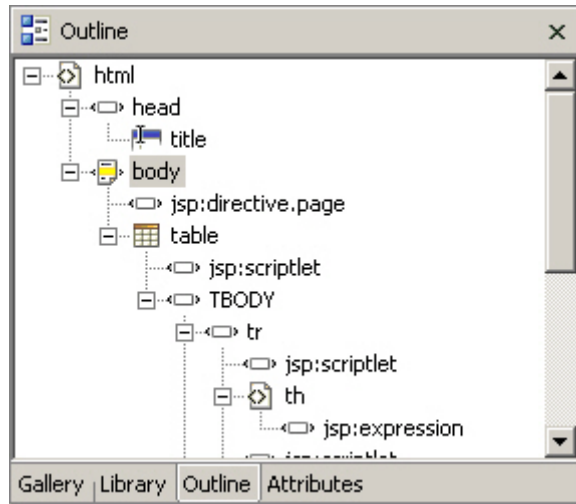
Behind the scenes, there's a lot more going on. To begin with, the JSP code editor is coupled with a parser that isolates the design logic from all the HTML and JSP tags and is displayed in the Design View.



The paradigm, previously presented that coupled the Visual Editor with source level changes, is now also paralleled here. You see the coupling of a source editor with tools to examine the underpinnings of the code components. We can take this to the next step.

### JSP Outline View

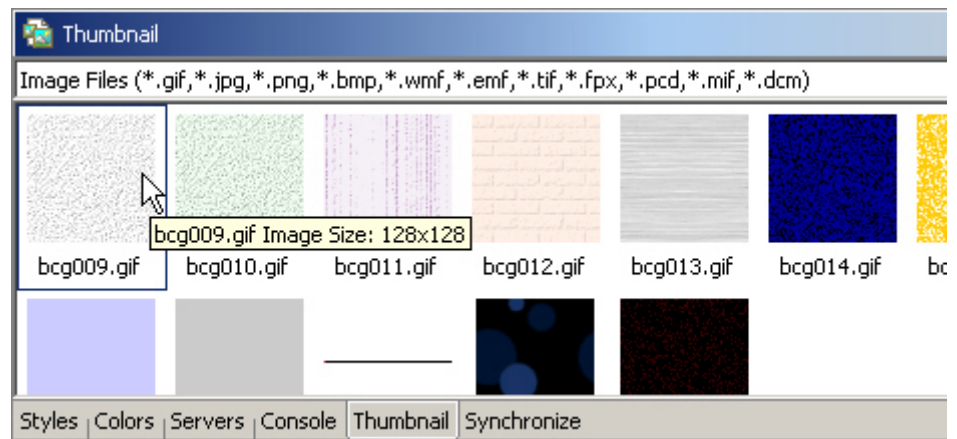
It's nice to be able to isolate the logic design. But what about the tag structure? There is an outline view that provides this.



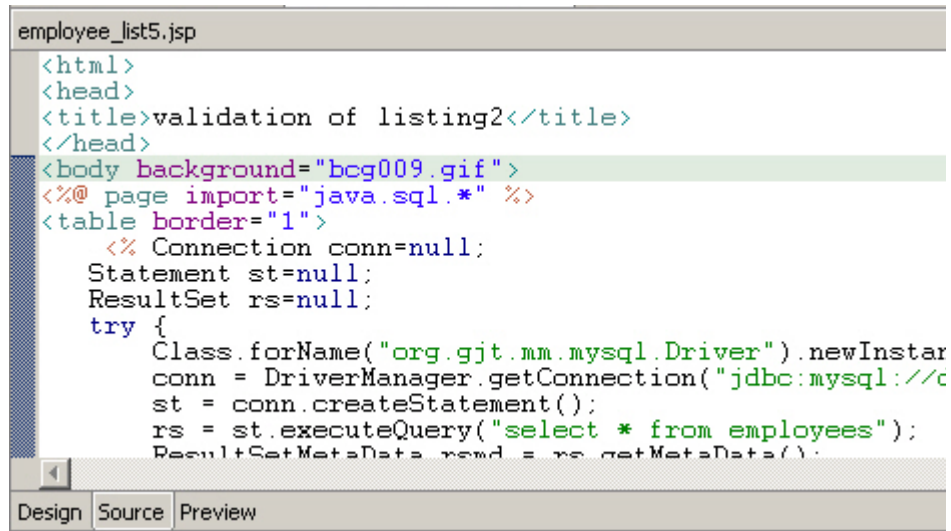
Moreover, drill down the tree as far as you need to go. As you are drilling down, the corresponding portions in the outline view are highlighted in the source editor. Drag the tag to different locations in your outline view and the corresponding text is moved in the source editor. We can further switch to the gallery view and adjust the visual appearance of the selected tag.

### Gallery Effects for the JSP

Clicking on the Wallpaper icon displays a catalog of swatches which you can associate with your JSP code.



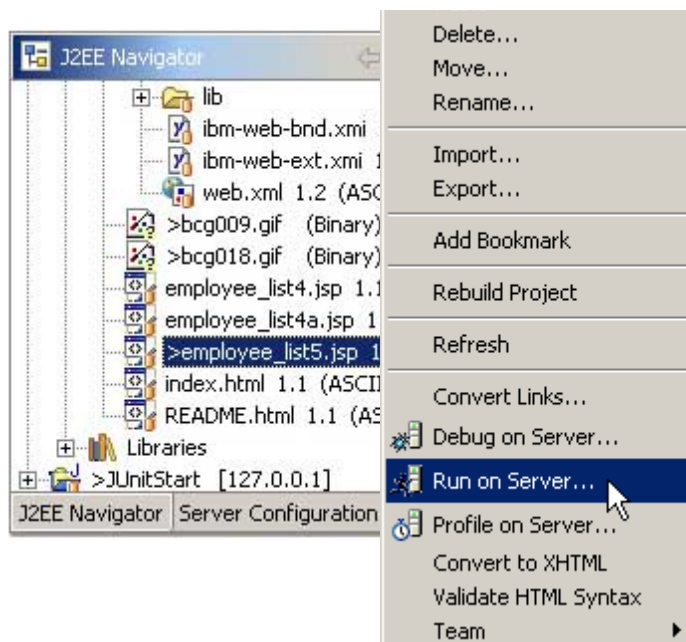
Double click on the desired pattern and the code is instantly inserted to your JSP.



```
employee_list5.jsp
<html>
<head>
<title>validation of listing2</title>
</head>
<body background="bcg009.gif">
<%@ page import="java.sql.*" %>
<table border="1">
<% Connection conn=null;
Statement st=null;
ResultSet rs=null;
try {
Class.forName("org.gjt.mm.mysql.Driver").newInstance();
conn = DriverManager.getConnection("jdbc:mysql://c
st = conn.createStatement();
rs = st.executeQuery("select * from employees");
ResultSetMetaData rowd = rs.getMetaData();
```

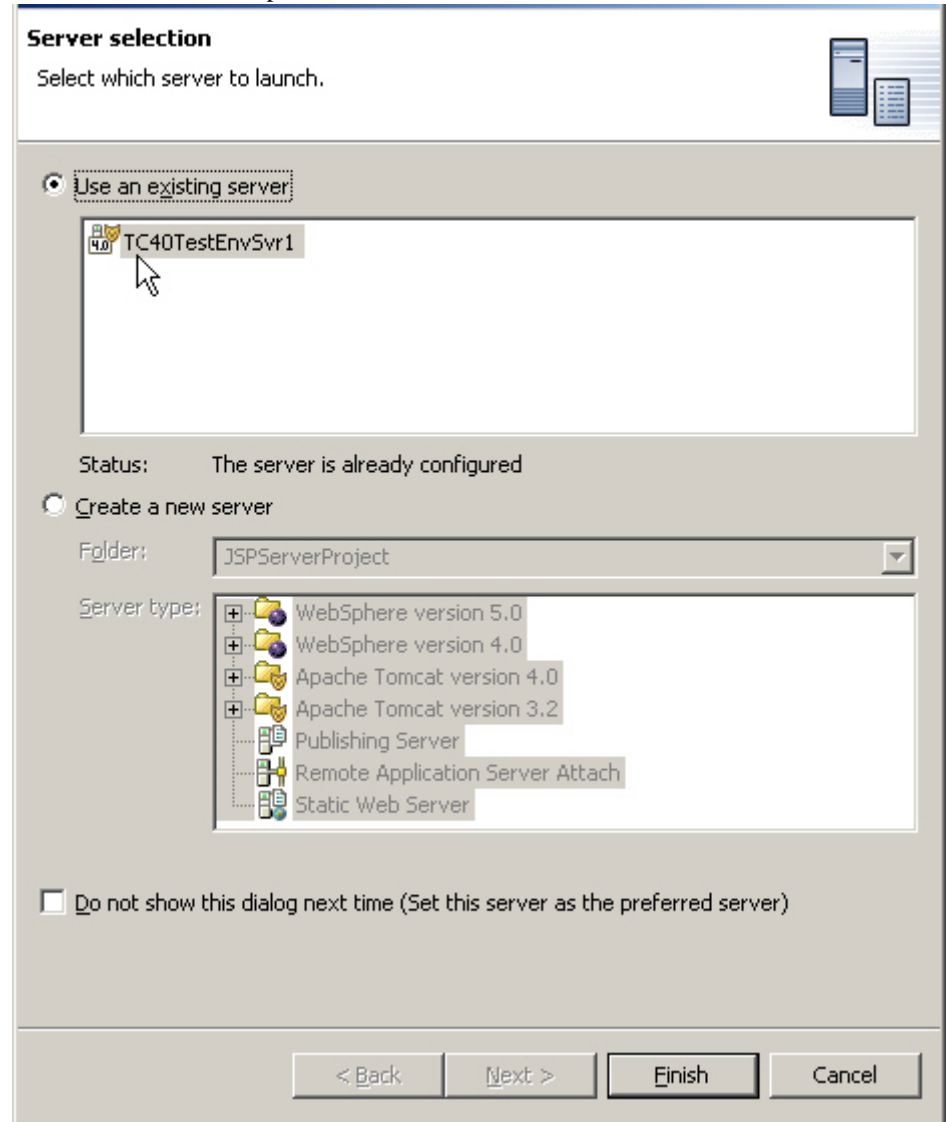
### Testing on the Application Server

The next step is to test drive your code by running it on an Application Server. Your workbench has a J2EE navigator. Right mouse-click on the name of your JSP Code and select the option to run the server.



**Choose from a selection of Servers (or define your own)**

You have a number of options available.



Aside from selecting/defining server configurations, you can set the debugging mode. .

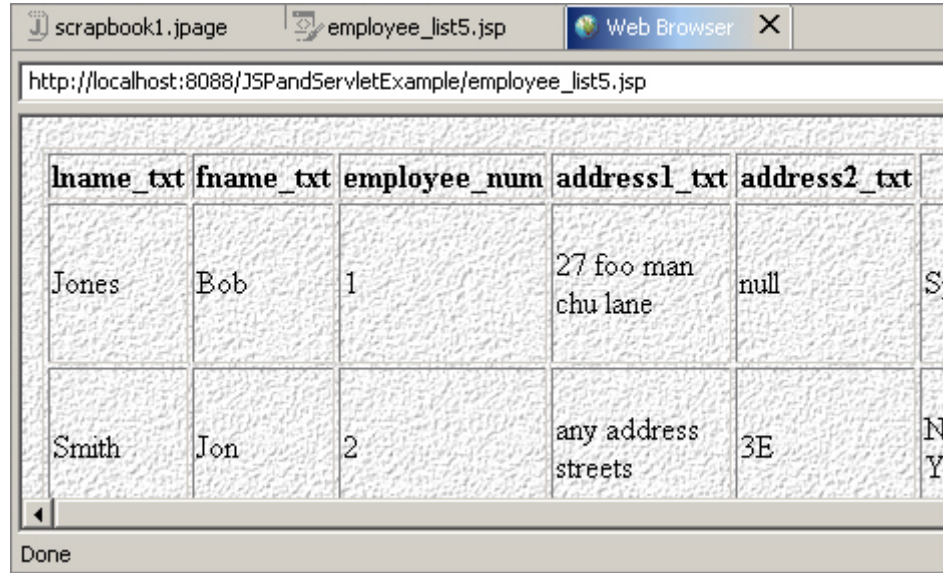
**TABLE 2.**

Representative Development and Deployment Options

Development Stage	Debug	Security
Development	ON	OFF
QA Testing	ON	ON
Production	OFF	ON

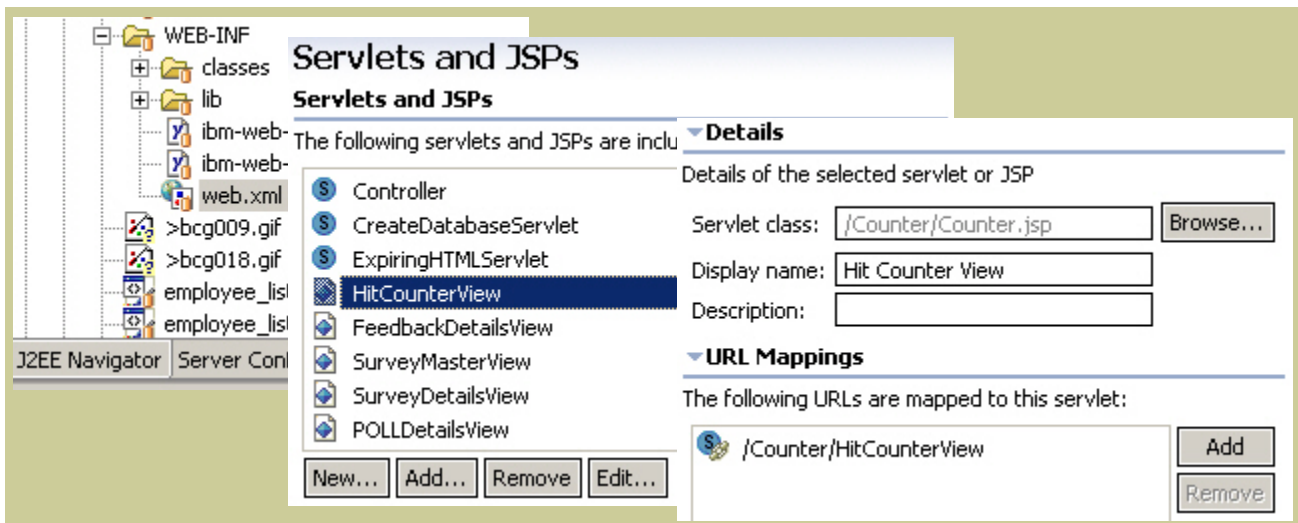
### The Completed JSP

The results of running the JSP are now shown in an embedded test browser.



### Productivity, Productivity and More Productivity

Often, developers generate tons of code and it is easy to lose track of where you are. Go to your deployment descriptor (web.xml) and inspect it.

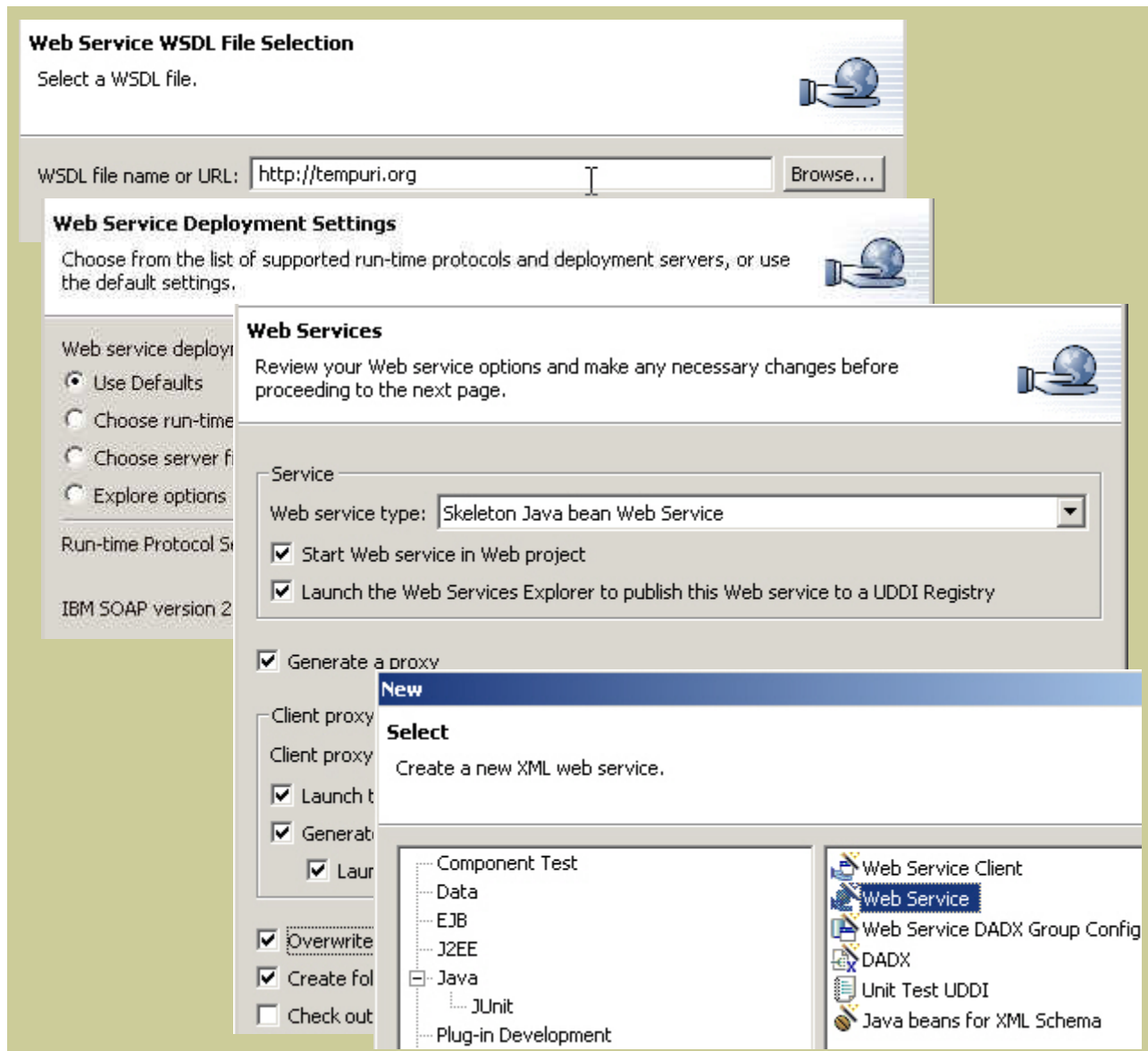


Like everything else in WebSphere Studio, it's all connected.



## 2.1 Web Services

Web Services is a very powerful technology. However, it is, to say the least, time consuming to keep up with all the new evolving facilities and low level code specifics, be it in the construction of a SOAP message or preparing a WSDL file. The preferable way of developing Web Services applications is to step through a wizard that does all the hard work for you; freeing you to concentrate on coding the methods that implement the services you are publishing.



### 3 Observations and Conclusions

---

Every company has its own set of unique development requirements and constraints. They all need to be able to achieve control over life cycle management along with cost containment. Sometimes, companies are not free to dictate which technologies they use. They may inherit specific technologies as a result of a merger or acquisition. Moreover, it is important in enterprise development to be able to mix and match components and tools on an as needed basis.

WebSphere Studio was designed to be an open framework. Strategy is to provide an open system while providing basic combination of features that make sense but allow the customer freedom to restructure this in any way appropriate.