

**Chapter  
1**

**GETTING  
STARTED**

*Get on the  
Fast Track!*



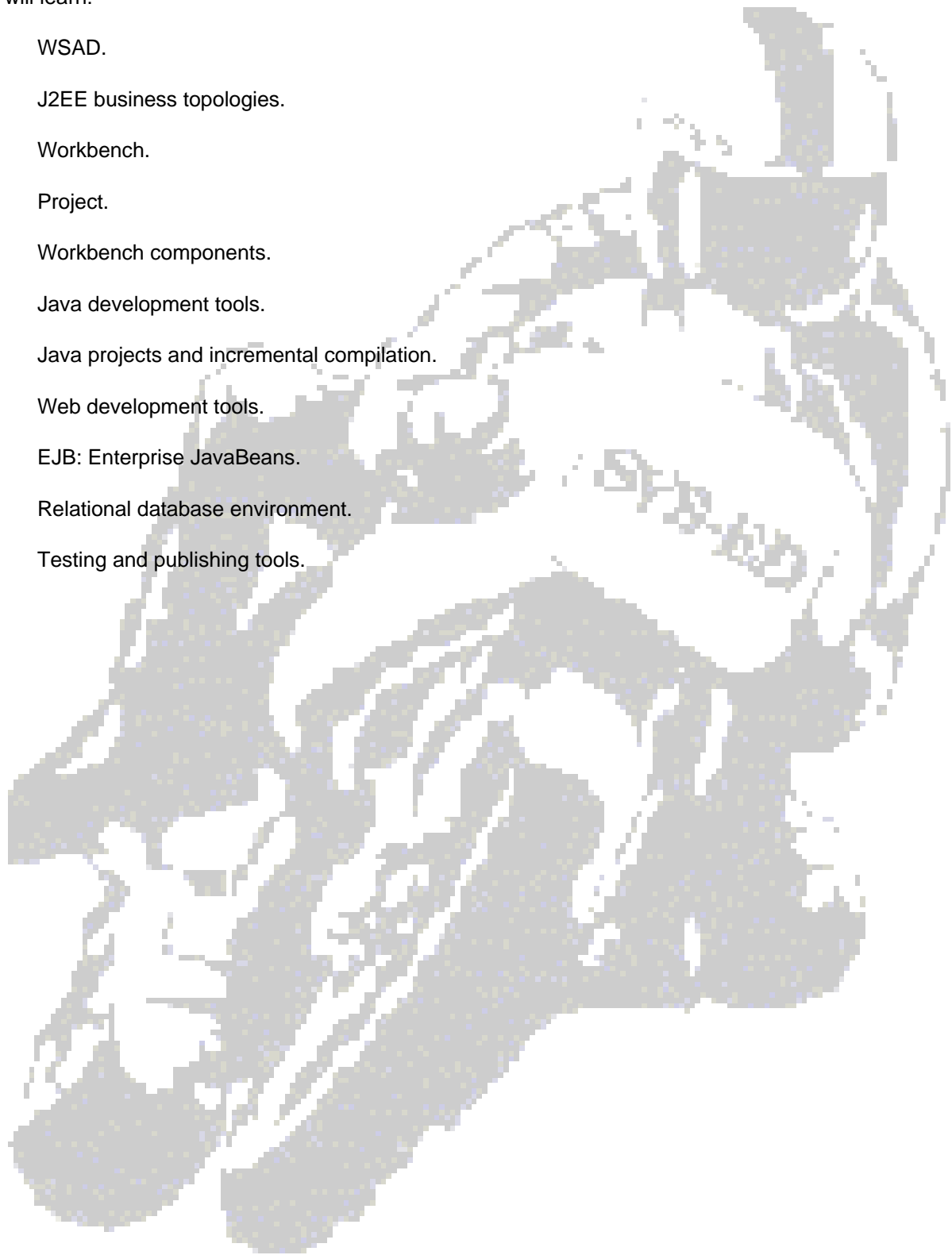
TM

**SYS-ED/  
Computer  
Education  
Techniques, Inc.**

## Objectives

You will learn:

- WSAD.
- J2EE business topologies.
- Workbench.
- Project.
- Workbench components.
- Java development tools.
- Java projects and incremental compilation.
- Web development tools.
- EJB: Enterprise JavaBeans.
- Relational database environment.
- Testing and publishing tools.



## **1 WSAD**

IBM has combined WebSphere Studio, Professional and Advanced Editions and VisualAge for Java Enterprise Edition into WebSphere Studio Application Developer - WSAD. Java is used as a server-side technology for processing Internet-based requests and securing access to corporate data.

Fortune 1000 corporations and government municipalities are utilizing Enterprise Java for implementing new systems. Most of this technology is based on the J2EE architecture, including JDBC, JSPs and Servlets. Enterprise JavaBeans and Servlets and Java Server Pages provide the capability to implement scalable distributed applications.

WSAD provides the capability to create J2EE systems and code, test and deploy the components. Wizards are provided for generating XML and skeleton code.

---

### **1.1 Open Standards**

Eclipse is a universal tool platform with extensible IDE. WebSphere Studio Application Developer is built on open standards and Eclipse. WASD provides portability.

Applications that are developed with Java 2 Enterprise Edition specifications provide developers with the capability for building server-side applications with the following standards:

- Servlets
- JSP
- EJB

WSAD can be used for changing the Java run-time environment - JRE. The JRE can be specified at the time a project is compiled or run.

WSAD supports a several source code repositories and supports the open software Concurrent Versions System - CVS.

## 1.2 Programming Productivity

WSAD provides the application developer with wizards, tools, and utilities to build applications.

The tools provided include:

- Automatic updates to web links.
- Graphics tools and generated JavaScript code.
- Refactoring utility to reorganize classes.
- A unit-test environment and debugger.
- Profiling tools for analyzing performance problems.



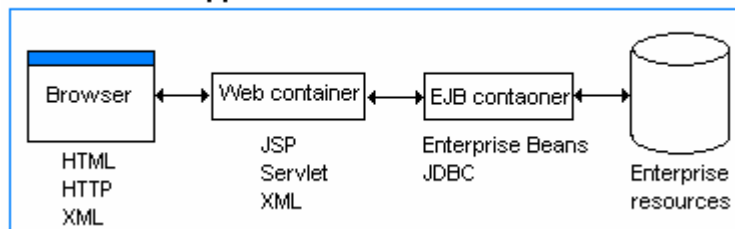
## 2 J2EE Business Topologies

A number of applications types can be developed using WSAD.

### 2.1 Multi-tier EJB Applications

EJB: Enterprise JavaBeans server-based components are reusable and implement business functionality.

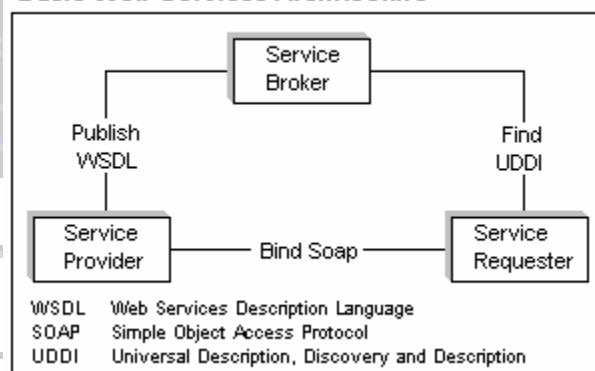
**EJB multi-tier application**



### 2.2 B2B Web Services Applications

WSAD provides for the creation and deployment of web services. A web service is a self-contained, modular application(s) that can be described, published, located, and invoked over the Internet.

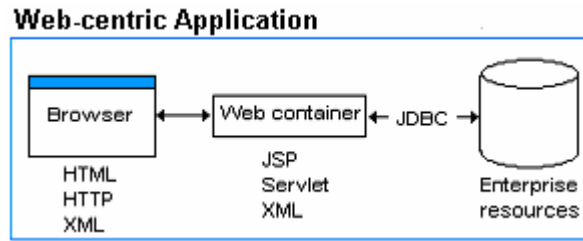
**Basic Web Services Architecture**



## 2.3 Web-centric | Servlet/JSP Applications

JSP: Java Server Pages extends web server functionality. JSP allows programmers to create dynamic Web content.

- JSP can be used to develop web applications that are server and platform-independent.
- Servlets are applications written in JAVA that run on the server.

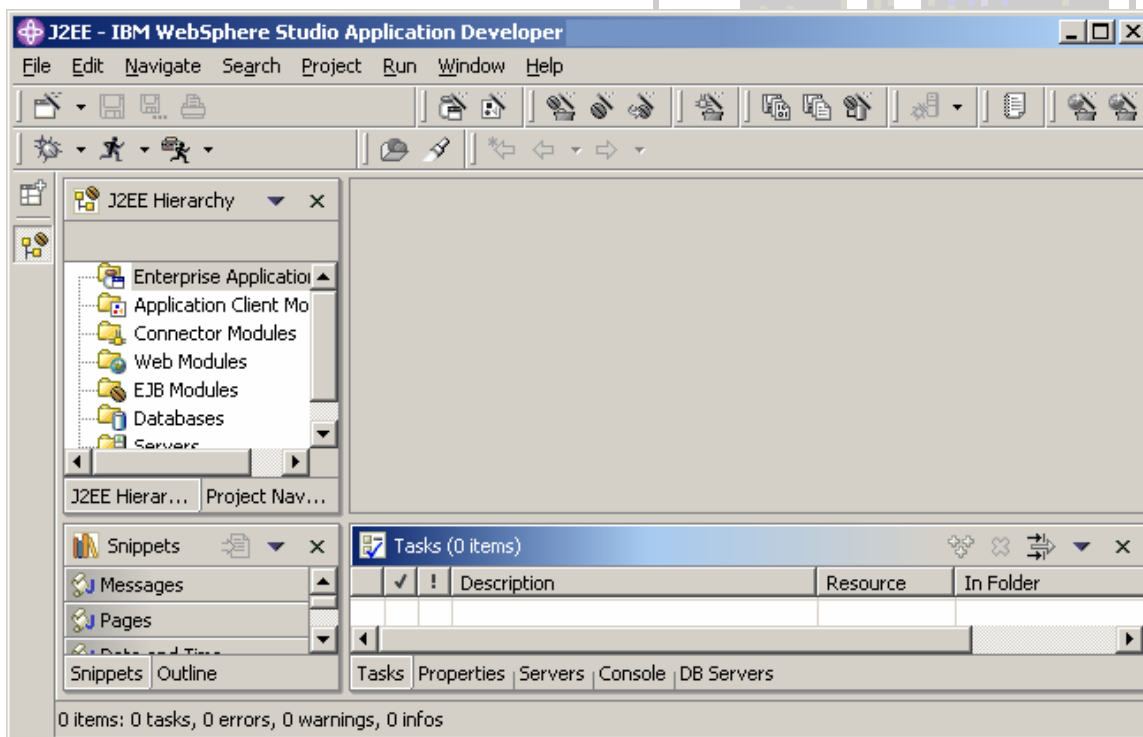


### 3 Workbench

The workbench is the user interface for WebSphere Studio. It is an integrated development environment and contains customizable perspectives that are combinations of views and editors.

The workbench provides a common way for the members of a project team to create, manage, and navigate resources. It contains a number of standard perspectives.

Views	Different ways of looking at the available resources.
Editors	Software to create and modify code.
Perspectives	Combinations of views and editors.



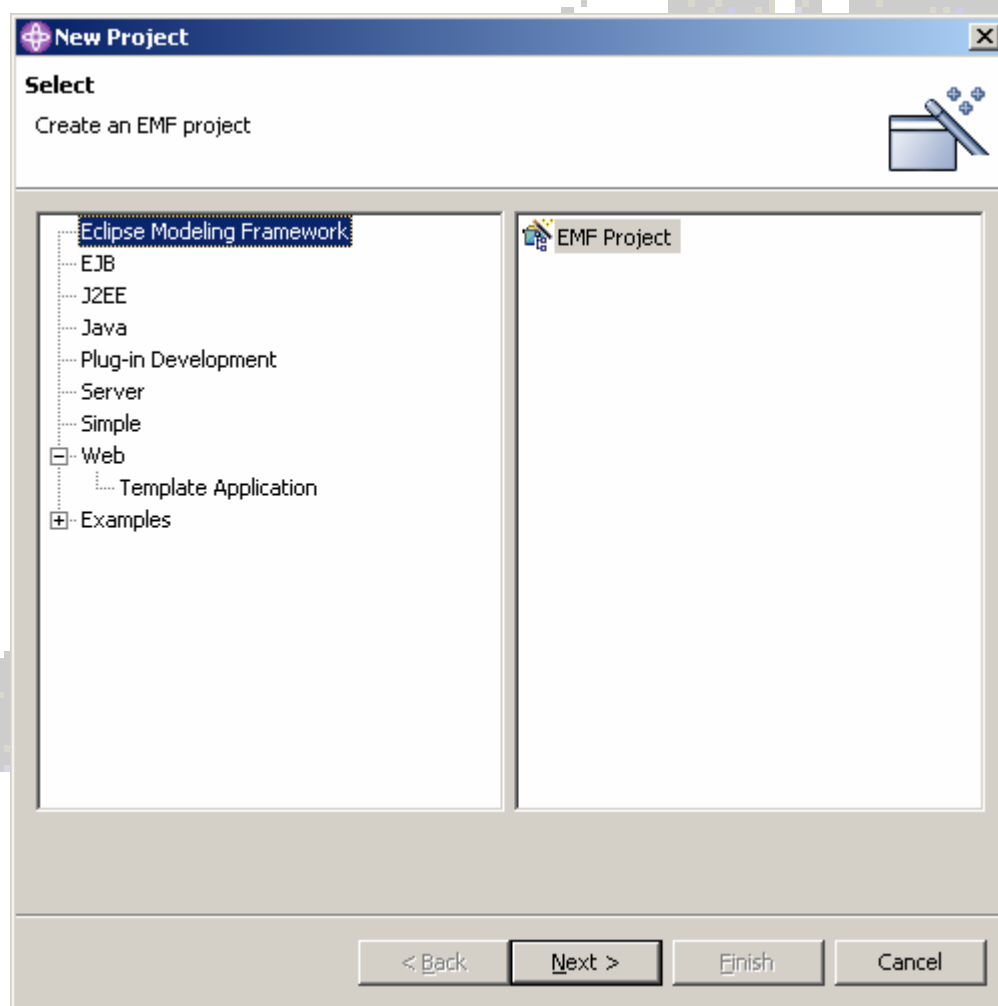
## 4 Project

A project is a group of directories and files that make up an application or part of an application. A project contains buildable units that can be compiled and executed. While working in the WSAD workbench, all files should be stored in project.

There are several different types of projects that can be created in WSAD including J2EE, EJB, Java, Server and Web projects.

There are predefined project resource mappings that control how the resources are mapped to files. Project build specifications control how it will be built, compiled, stored, and retrieved.

A project is either open or closed. When a project is closed, it cannot be changed in the workbench. This screen capture illustrates the beginning of a project creation.



---

## 4.1 Project Types

The following types of projects can be created for developing J2EE applications:

Type of Project	Description
Enterprise Application	Contains Web modules, EJB modules, JAR files, and application client modules to create a complete enterprise application.
Web	Servlets, JSPs, Java files, or static documents to create a web application.
EJB	Resources for an EJB applications.
Application Client	Resources needed for application client modules.
Java	Java code for a Java application.
Server	The test environment for applications.
Simple	Generic projects with source code.

---

## **5 Workbench Components**

---

### **5.1 Perspectives**

The perspective is a group of views and editors for performing a set of development activities. Multiple perspectives can be open in a single workbench window.

It is possible to jump from perspective to perspective in the same window. A perspective can be opened in new workbench windows.

New perspectives can be created by the developer. Existing perspective can be configured by the programmer.

---

### **5.2 Views**

These are different windows that present navigational information; this set of windows is called a view.

One of the most commonly used views is the Navigator. The Navigator depicts project, directories in the projects and files in a hierarchy. A view can be a window or stacked with other views.

---

### **5.3 Editors**

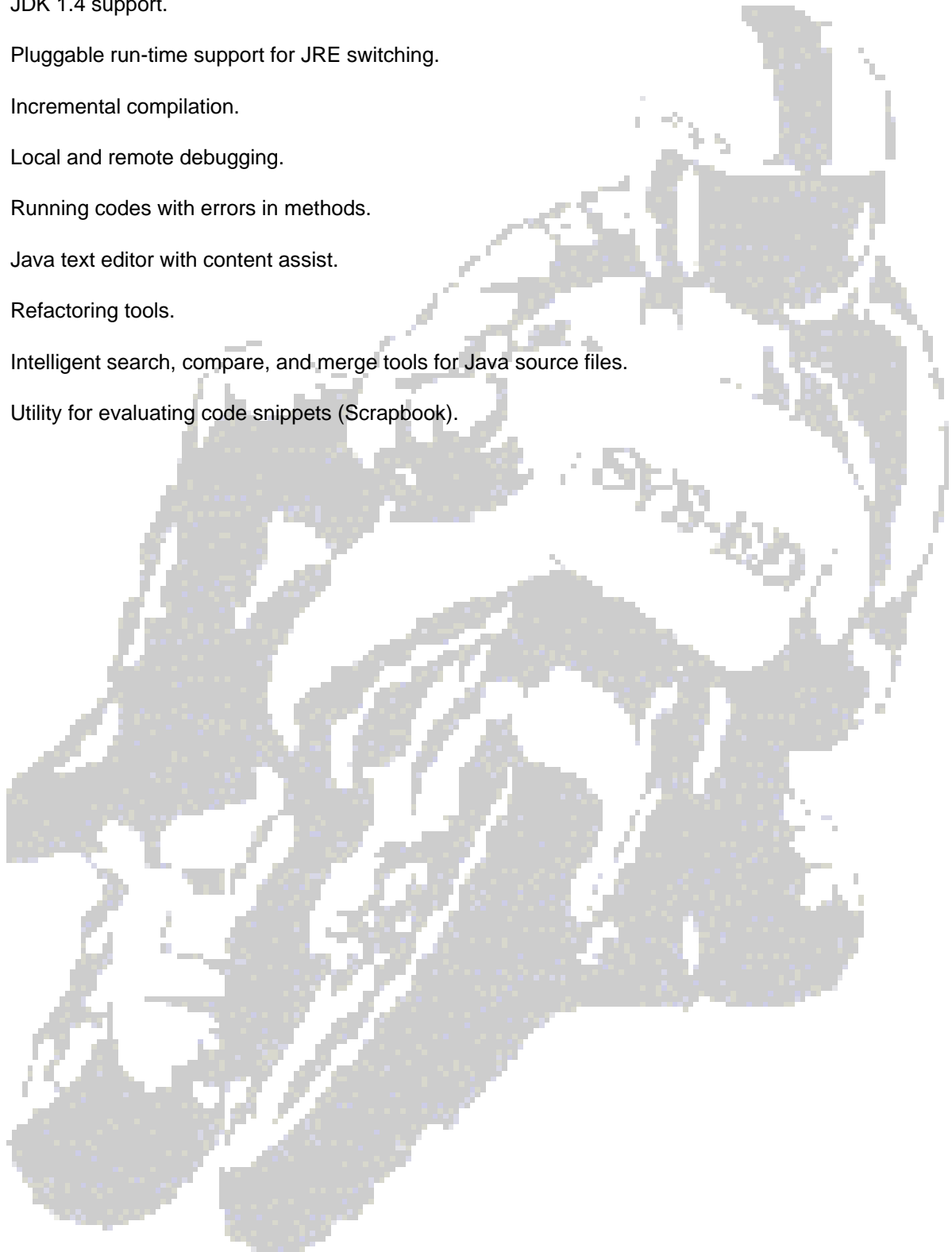
Based on the file extension, an associated editor is executed. This is determined by the workbench when a user clicks or opens a file. The association is configurable.

---

## **6 Java Development Tools**

The Java development tools offer the following capabilities:

- JDK 1.4 support.
- Pluggable run-time support for JRE switching.
- Incremental compilation.
- Local and remote debugging.
- Running codes with errors in methods.
- Java text editor with content assist.
- Refactoring tools.
- Intelligent search, compare, and merge tools for Java source files.
- Utility for evaluating code snippets (Scrapbook).



---

## **7 Java Projects and Incremental Compilation**

WSAD generates a Java project structure. It includes a Java builder that can incrementally compile Java source files as they are changed.

The project can be used as the source container. All source and packages are inside the project. The class files are also stored in the project.

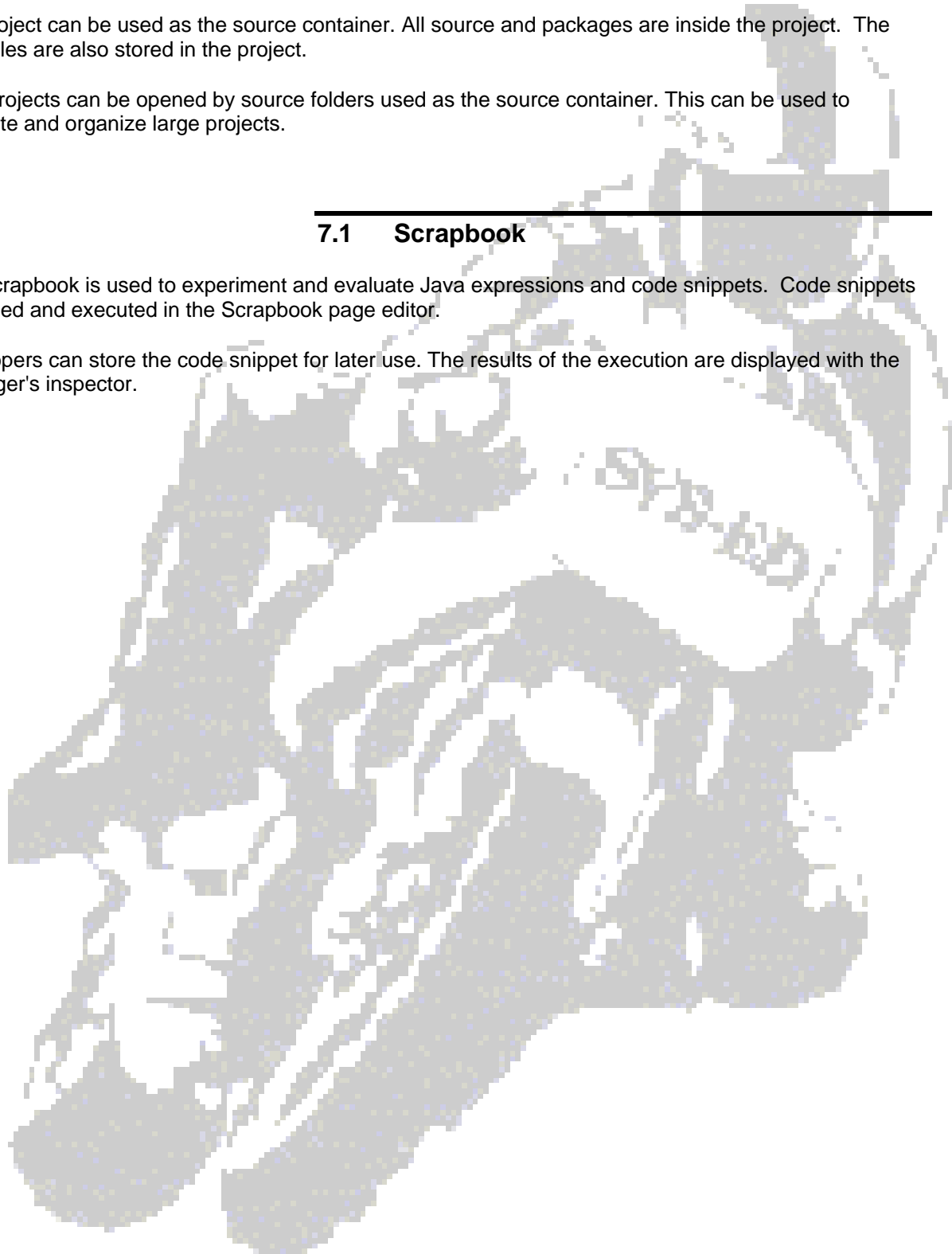
Java projects can be opened by source folders used as the source container. This can be used to separate and organize large projects.

---

### **7.1 Scrapbook**

The Scrapbook is used to experiment and evaluate Java expressions and code snippets. Code snippets are typed and executed in the Scrapbook page editor.

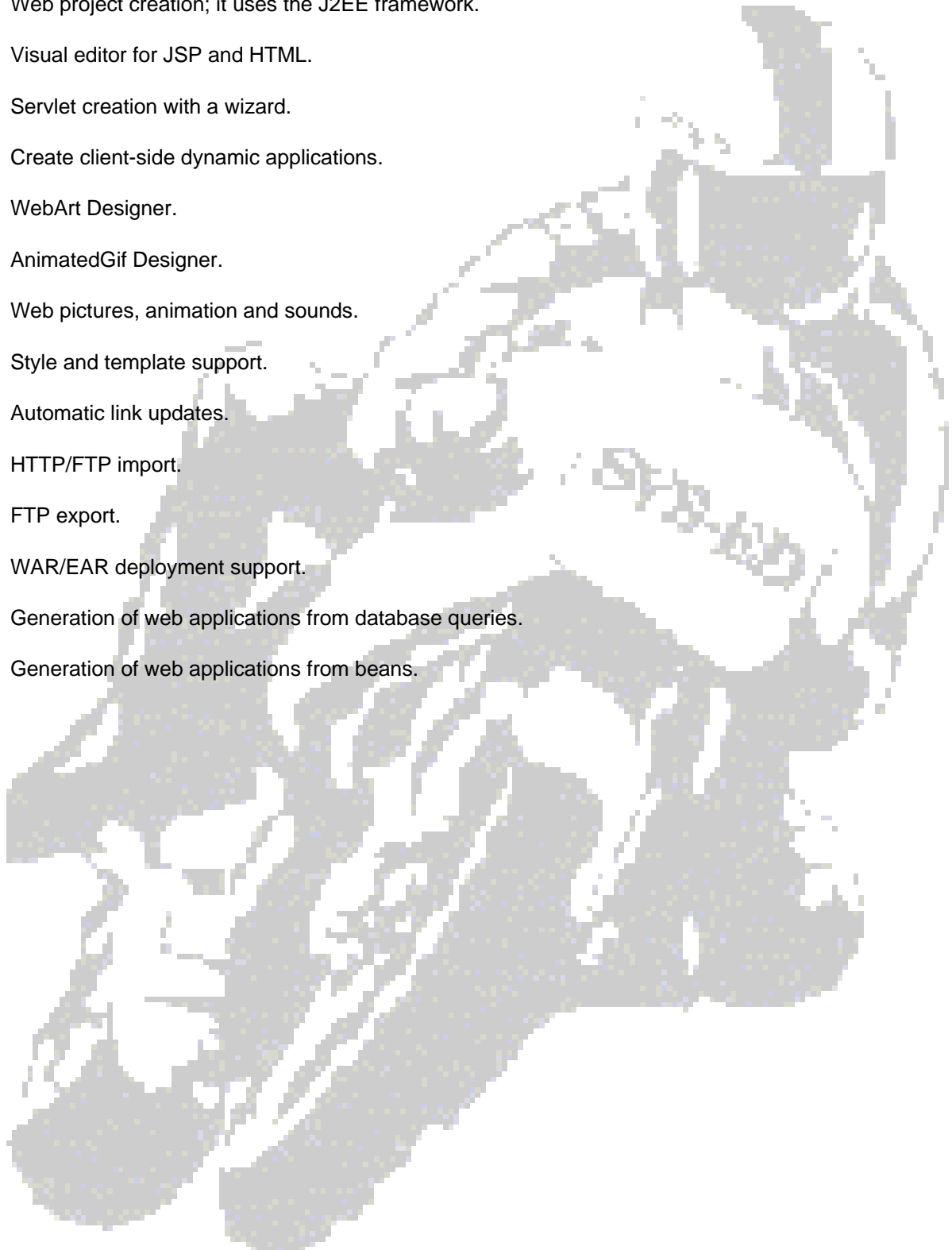
Developers can store the code snippet for later use. The results of the execution are displayed with the debugger's inspector.



## 8 Web Development Tools

The web development environment provides the following features:

- Web project creation; it uses the J2EE framework.
- Visual editor for JSP and HTML.
- Servlet creation with a wizard.
- Create client-side dynamic applications.
- WebArt Designer.
- AnimatedGif Designer.
- Web pictures, animation and sounds.
- Style and template support.
- Automatic link updates.
- HTTP/FTP import.
- FTP export.
- WAR/EAR deployment support.
- Generation of web applications from database queries.
- Generation of web applications from beans.



## 9 EJB: Enterprise JavaBeans

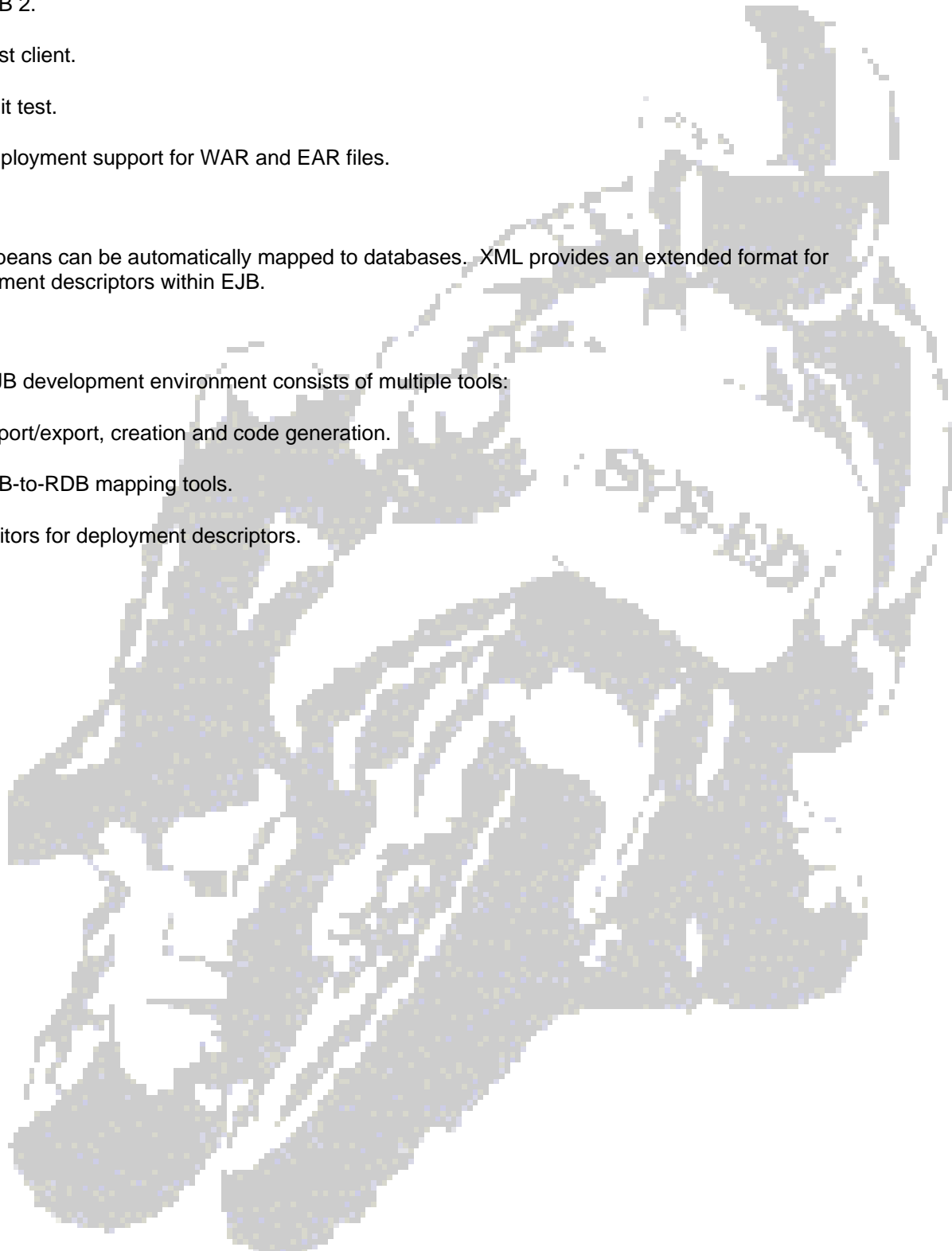
The features of the WSAD EJB development include:

- EJB 2.
- Test client.
- Unit test.
- Deployment support for WAR and EAR files.

Entity beans can be automatically mapped to databases. XML provides an extended format for deployment descriptors within EJB.

The EJB development environment consists of multiple tools:

- Import/export, creation and code generation.
- EJB-to-RDB mapping tools.
- Editors for deployment descriptors.



---

## **10 Relational Database Environment**

The WSAD Data perspective permits the user to create and view the data design. It will look like relational database schemas. A developer can browse or import database schemas.

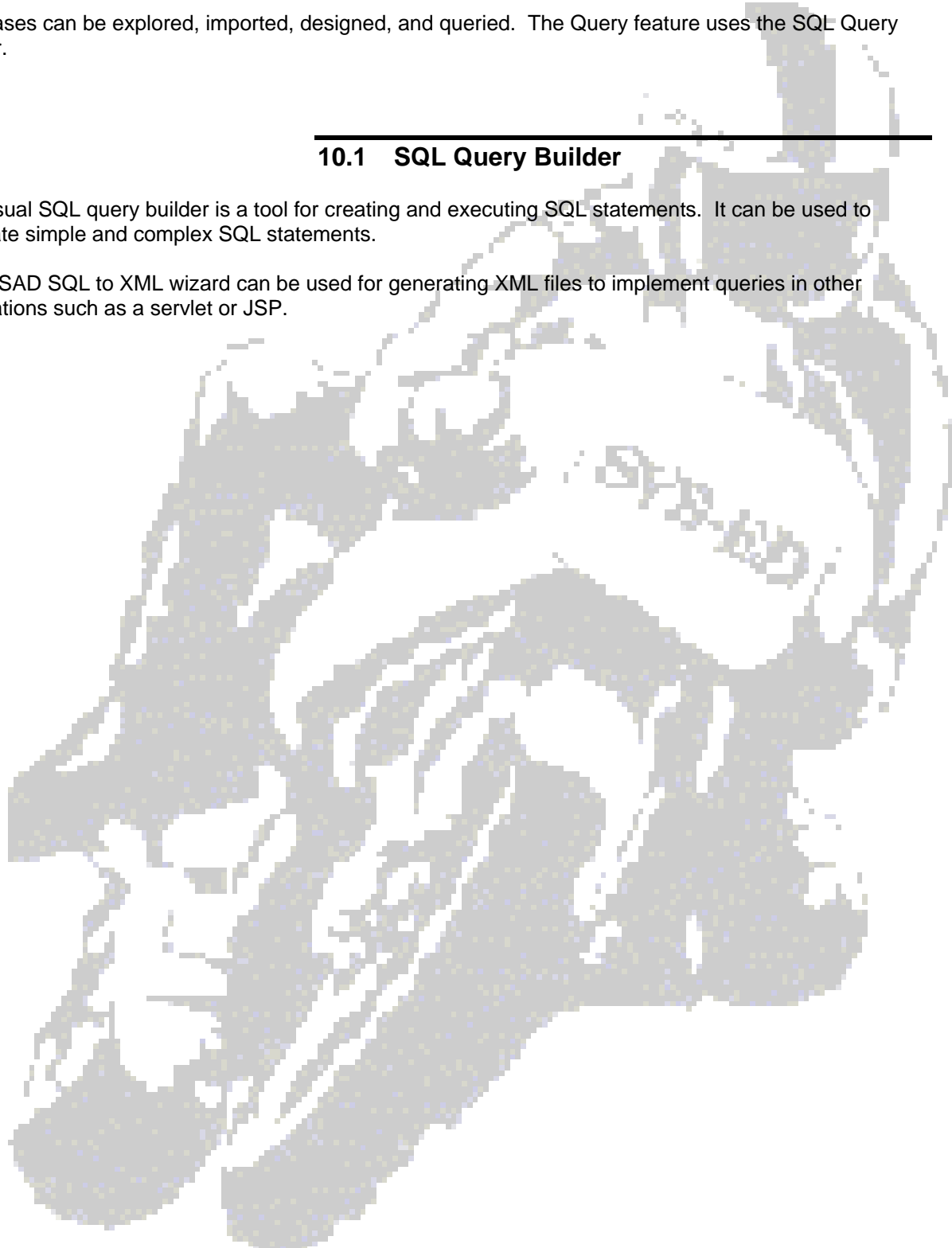
Databases can be explored, imported, designed, and queried. The Query feature uses the SQL Query Builder.

---

### **10.1 SQL Query Builder**

The visual SQL query builder is a tool for creating and executing SQL statements. It can be used to generate simple and complex SQL statements.

The WSAD SQL to XML wizard can be used for generating XML files to implement queries in other applications such as a servlet or JSP.



---

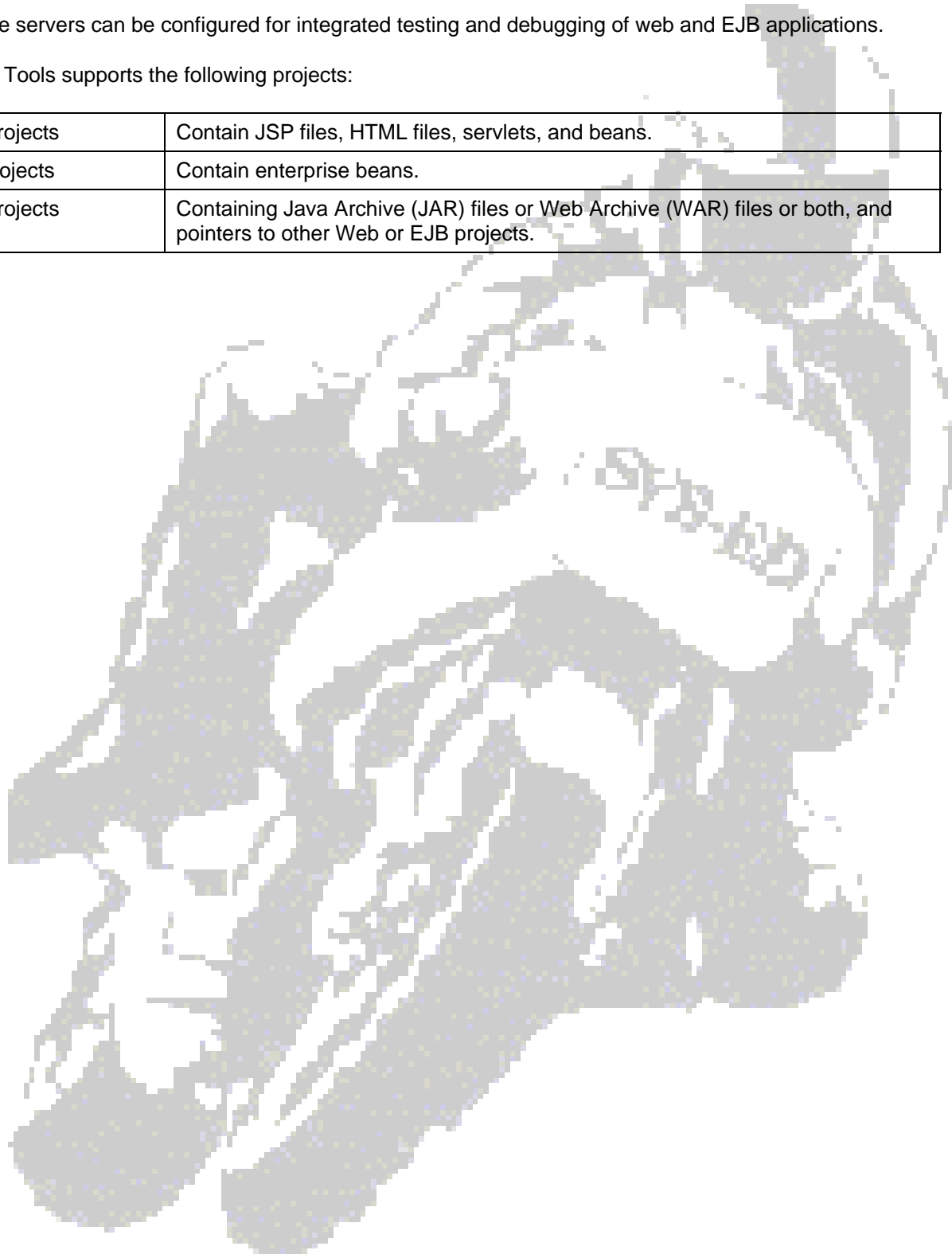
## 11 Testing and Publishing Tools

The Server Tools feature provides a unit test environment where JSP files, servlets, and HTML files can be tested.

Remote servers can be configured for integrated testing and debugging of web and EJB applications.

Server Tools supports the following projects:

Web projects	Contain JSP files, HTML files, servlets, and beans.
EJB projects	Contain enterprise beans.
EAR projects	Containing Java Archive (JAR) files or Web Archive (WAR) files or both, and pointers to other Web or EJB projects.



---

## 12 Profiling and Logging Tools

---

### 12.1 Performance Profiling

WebSphere Studio provides tools for testing an application's performance early in the development cycle. Architectural changes can then be made to reduce risk and avoid problems in the final performance tests.

---

### 12.2 Graphical and Statistical Profiling Views

The Profiling tools collect data on a Java program's run-time behavior, and present this data in graphical and statistical views. In addition memory leak diagnostic aids, there are statistical views which afford pattern extraction and information exploration capabilities.

---

### 12.3 Sequence Diagram Views

Sequence Diagram views provide a means to visualize the execution of Java applications in the form of sequence diagrams that have been defined by the UML: Unified Modeling Language notation.

These views are used for following purposes:

User View	Purpose
Software development management personnel.	Verifying a software implementation with the design documentation for the purpose of accuracy.
Software development management personnel.	Understanding the execution flow of an application and Analyzing its performance.
Business application administrators.	Monitoring the state and performance of such applications, where the monitoring may be performed at any level: <ul style="list-style-type: none"><li>• agent</li><li>• process</li><li>• host</li><li>• monitor - cluster of hosts.</li></ul>
Computer network administrators.	Monitoring interactions among hosts on the network.

---

## 12.4 Distributed Process Monitoring

The Profiling tools are also used to launch remote applications. Multiple, live processes residing on different machines can be monitored concurrently.

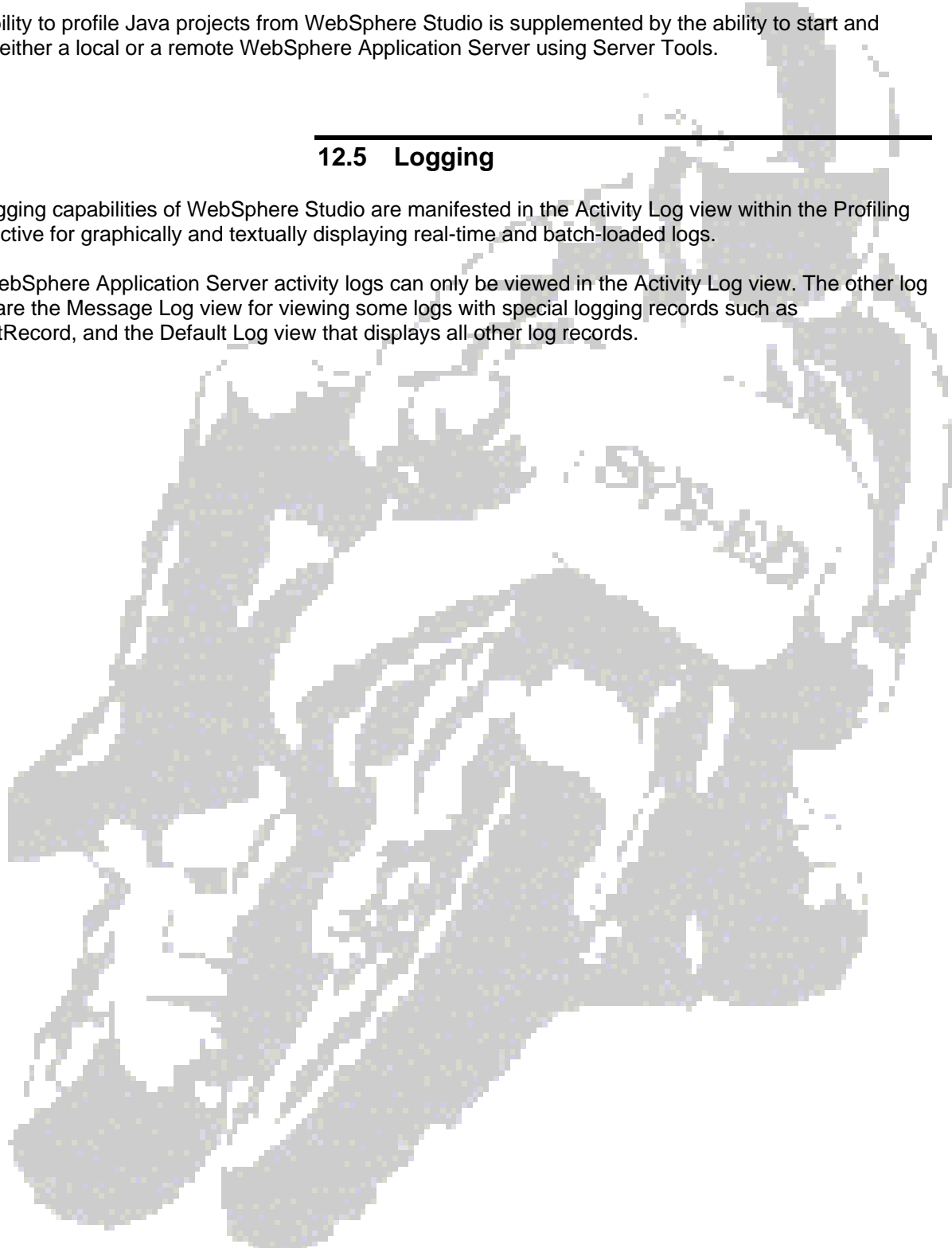
The ability to profile Java projects from WebSphere Studio is supplemented by the ability to start and profile either a local or a remote WebSphere Application Server using Server Tools.

---

## 12.5 Logging

The logging capabilities of WebSphere Studio are manifested in the Activity Log view within the Profiling perspective for graphically and textually displaying real-time and batch-loaded logs.

The WebSphere Application Server activity logs can only be viewed in the Activity Log view. The other log views are the Message Log view for viewing some logs with special logging records such as DefaultRecord, and the Default Log view that displays all other log records.



## 13 Debuggers

All products based on Eclipse include a debugger that can be used to detect and diagnose errors in the programs running either locally or remotely. The debugger can be used to control the execution of programs by setting breakpoints, suspending execution, stepping through the code, and examining the values of variables.

Both, live server-side code and programs running locally on a workstation can be debugged.

The debugger includes:

Debug view	Shows threads and stack frames.
Processes view	Shows all currently running and recently terminated processes.
Console view	Used to interact with running processes.

There are also views that display breakpoints and let the programmer inspect variables.

The following debug components can be used for debugging locally and remotely:

WebSphere Application Server debug adapter	Used to debug web objects that are running on WebSphere Application Server. These objects include EJBs, JSPs, and servlets. It can be also used to debug server-side JavaScript.
Active Script debugger	Used to detect and diagnose errors in code that contains Active Script or Active Script objects.
Java development tools (JDT) debugger	Used to debug Java.
SQL Stored Procedure Debugger	Used to detect and diagnose errors in SQL procedures that are running on a DB2 server.
SQLJ debugger	Used to debug Java embedded with SQL.
Compiled language debugger	Used to debug compiled-language applications.

Hot-method replace is used to modify source code in a method when an application is stopped on a break point. The code can be modified and restarted without having to restart the server for the change to be recognized.