

Scope and coverage of the Certification Guide

Chapter topics

- ❖ *Technology and tools*
- ❖ *WebSphere Studio*
- ❖ *WebSphere application development*
- ❖ *A dual overview of the guide*

Certification objectives

- ❖ *This chapter does not address any specific certification objectives*

Welcome to the Certification Study Guide for *An Introduction to Web Application Development with IBM WebSphere Studio*. This chapter will provide you with a context for the rest of the Certification Guide.

Technology and tools

Certifications are associated with roles. The role to which this Certification Guide applies is that of IBM Certified Associate Developer—WebSphere Studio, V5.0: that is, for new developers to the Java Platform 2, Enterprise Edition (J2EE). A person having this role provides application development services related to design, implementation, debugging, and deployment of J2EE Web applications. The specific technology components include servlets, JavaServer Pages (JSP), JavaBeans, HTML, and JDBC. Prerequisite to mastering these technologies is a background in application development using Java and the corresponding skills in building object-oriented software artifacts.

Note: For complete descriptions of all available Certification tests from IBM, see www.ibm.com/certify. For sample tests see certify.torolab.ibm.com.

J2EE is a very large technology domain. Most of the enterprise Application Programming Interfaces (APIs) associated with J2EE are *not* required by this certification role nor covered by this certification guide.

A second focus of the certified role is the platform and tools used during the development tasks. In particular the IBM Certified Associate Developer—WebSphere Studio, V5.0 role is targeted to developers who are new adopters of WebSphere Application Development technology, namely WebSphere Studio products. Thus, the focus of this guide is to highlight the developer tasks performed using the WebSphere Studio application development platform.

WebSphere Studio

The WebSphere Studio product family is a collection of tools built on an open application development framework. This base, called the WebSphere Workbench, is a robust offering by IBM of the Eclipse framework (see www.eclipse.org). This framework allows application development tools to be *plugged in* in a very dynamic fashion. This allows the product to grow and evolve both with a development team and as technology changes.

WebSphere Studio (and Eclipse) provides a role-based organization to the Integrated Development Environment (IDE). There are distinct *perspectives* organized around the specific tools and views need to perform a specific task or series of tasks. Frequently these perspectives are structured around the type of artifacts that are being developed or tested, such as Java code, Web assets, XML documents, data access, J2EE components, and debugging. Developers can easily adapt and create their own perspectives to reflect the tasks that they perform and the way in which they perform them.

With functionality layered and plugged in, it is possible for IBM to provide different entry points for customers to purchase and use WebSphere Studio. That is why IBM currently ships a number of different products under the WebSphere Studio name (www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/studio). The principal editions of interest to this audience are the following:

- WebSphere Studio Site Developer
- WebSphere Studio Application Developer
- WebSphere Studio Application Developer – Integration Edition
- WebSphere Studio Enterprise Edition

Each of these products has an increasing set of features. For this certification guide, all of the role-specific tasks we will be concerned with are supported by any of these editions. As you will come to appreciate through working with WebSphere Studio, this product family is a very powerful and productive development environment. For a more complete description of WebSphere Studio, check out Chapter 3. In addition, you can visit one of the Web resources listed at the end of Chapter 3.

This book ships with a copy of WebSphere Application Server – Express. This offering supplies a copy of WebSphere Studio Site Developer. This will be the edition that most of you will use to work through the exercises in the book. However, the look and feel and operations will be the same in any edition you choose to work with.

Web application development

So if the role we are targeting is involved in Web application development, what is a Web application? A Web application is server-side software that is Web enabled. In our case, Web applications refer to implementations that are accessed via the Web through a J2EE-compliant application server. As you will see in Chapter 7, the Web gateway component is a Java servlet. This is a server-side Java (J2EE) component that enables a Web (HTTP) request to be serviced by Java code running on an application server.

Java is a very powerful development platform (much more than a programming language). With the wide range of API (class libraries) available to a Java developer, the entire range of enterprise resources is available to be manipulated. Thus J2EE, and servlets in particular (along with various connector APIs), provide a rich foundation on which to Web-enable your enterprise.

Direct Web access to enterprise data and function is not likely to be attractive to information technology (IT) departments because of the obvious issues of security and data integrity that come into play. J2EE is particularly powerful in addressing those concerns. The scope of the Associate Developer role, and of its Certification and Certification Guide, is constrained to the entry-level aspects of J2EE technology. As with most things in life, you need to learn to walk before you can run. We aim to provide you with a solid foundation upon which you can start to train for a marathon in some distant future.

To appreciate the J2EE component model and the full enterprise APIs, you first need to understand the underpinnings. The J2EE Web components and JDBC are what provides that perfect entry platform. Many large IT shops that have been developing and deploying J2EE applications for many years develop Web components exclusively. So, although we are excluding a large part of J2EE (look at Tests 484 and 287 if you are interested in this more complete scope), you will still end up with a very powerful development model and deployment platform.

For a more concrete example, suppose you want to build a customer service site in which Web users can search through a problem resolution database. Such a set of integrated applications (services) can easily be written using the Java and J2EE technology presented in this Certification Guide.

Why combine a technology-based role with a specific set of application developer job tasks? Further, why target those job tasks to a particular development platform (WebSphere Studio)? Productivity! That is the simple answer. Any role-based certification tries to measure competency. In particular, it is concerned with measuring competency at performing specific tasks. Yes, I can write Java code using a simple text editor. But once my applications grow to any reasonable size, my ability to manage the components effectively and be able to maintain that code quickly (including adding new functionality) is greatly constrained by that tool set.

If, on the other hand, you or your organization chooses a development platform, then you can be productive in building applications using that development platform only if you are well versed in letting the tool(s) perform their assistive magic. A corollary to this concept is that it is important that you learn not to use the tool in a way that will cause you to be counterproductive—that is, don't fight the tool. What follows, then, is that being productive building a servlet or a JSP page implies being productive building these components using the targeted development platform, in our case WebSphere Studio.

Each of the chapters in this guide are targeted to arm you with the expertise to understand the technology components while mastering a subset of the capabilities of the WebSphere Studio family that support building, testing, debugging, and deploying those components.

A dual overview of the guide

There are two different ways to approach this Certification Guide. The first is considering the guide as a tutorial for building Web applications. For this purpose, the guide is organized in a logical progression of tools, technology, and techniques. The best overview is a quick survey of the chapters. The second approach is more certification centered. For the reader who is primarily preparing to take Test 285, we provide a cross reference to the Certification Test Objectives.

Overview of the chapters

Chapter 2, *System requirements and installation*: This chapter helps you install WebSphere Application Server – Express edition. On the CD (included with the book) is WebSphere Studio Site Developer. After completing this installation task you will be ready to start developing on the WebSphere Studio platform.

Chapter 3, *Overview of Eclipse, Studio Workbench, and Studio*: This is a very strong chapter that provides the reader with a good overview of WebSphere Studio in general and certain aspects of Site Developer in particular.

Chapter 4, *Developing a simple Java application*: The basis of much J2EE development in WebSphere Studio is the development of Java code assets. This chapter covers the Java development tools in WebSphere Studio.

Chapter 5, *Debugging techniques*: An integrated source code debugger is one of the most significant productivity tools in any IDE. The WebSphere Studio debugger is no exception. This chapter works at building your proficiency with debugging Java code in WebSphere Studio.

Chapter 6, *Essential HTTP and HTML*: Building Web applications involves building Web assets, e.g., HTML files and JSP files. To be effective you must have a good working knowledge of HTTP and HTML.

Chapter 7, *Introduction to servlets*: The heart of the Web application, servlets, are explored in this chapter.

Chapter 8, *Handling HTTP requests*: Most servlet processing comes down to working with the `HttpServletRequest` and `HttpServletResponse` objects. Much of the response object processing is covered in Chapter 7. Chapter 8 looks at processing incoming request parameters and having them affect business processing.

Chapter 9, *Case study*: This is an overview of the use cases associated with a simple purchase order/invoicing application used to illustrate concepts throughout the rest of the Certification Guide.

Chapter 10, *Application state with servlets*: A single servlet rarely represents an entire application. Therefore the server needs to remember the application state of a user that makes several HTTP requests in order to carry out the application behavior. This chapter looks at the `HttpSession` API to manage application state within the J2EE Web Container.

Chapter 11, *Model-View-Controller basics*: Application layering is an extremely important tool when building software. Web application development is no exception. Model-View-Controller (MVC) is a common framework or pattern for isolating distinct application layers and structuring your code. The result is assigning application components to the appropriate layer.

Chapter 12, *Introduction to JavaServer pages*: The chapter surveys the basics of JavaServer Pages (JSP). This is a powerful scripting technology for building dynamic presentation (View) logic.

Chapter 13, *Controller design principles*: With MVC and more involved application designs, error handling, exceptions, logging, and such issues need to be addressed in a thoughtful way. This chapter looks at several of these design principles.

Chapter 14, *JSP tag libraries*: One of the tools used to extend JSP is the use of Custom Tags. These can be used to greatly improve readability and maintainability (as well as reuse) in JSP. This chapter looks at the mechanism to define custom tags while focusing on standard tag libraries (JSTL).

Chapter 15, *Accessing databases with JDBC*: We can't do business logic without some sort of connectivity API. JDBC is a central component in building Web applications. This chapter shows how to use JDBC with J2EE data sources to access corporate data located in relational data stores.

Chapter 16, *Deploying applications*: Finally, you have built, tested, and debugged your applications (over and over) within the development environment. So now you want to deploy to WebSphere Express. This is the chapter for you.

If you are ready, go ahead and jump off to the point that seems most appropriate to you. If you are not immediately going to build code from the product on the accompanying CD, start with Chapter 3. If you already have some level of experience with basic J2EE or even WebSphere Studio, start wherever is appropriate.

Cross referencing certification objectives

Certification Test Objective	Corresponding Chapter(s)
1. WORKBENCH BASICS	
A. Create J2EE projects	Chapters 3, 7, and 12 (and others)
B. Set workbench preferences	Chapter 3
C. Work with perspectives	Chapter 3
D. Import to and export from the workbench	Chapter 3 (and various)
E. Use the Help feature to aid in development activities	Chapter 3
2. JAVA DEVELOPMENT	
A. Create Java project, packages, classes, methods	Chapters 3 and 4
B. Use outline view	Chapters 4 and 6
C. Use Java browsing view	Chapter 4
D. Use refactoring features	Chapters 3, 4, and 11

E. Use content assist function	Chapters 3, 4, and 12
F. Add and organize import declarations	Chapter 4
G. Use search function	Various
H. Use task view	Chapter 4
I. Use hierarchy view	Chapter 4
J. Use property dialogs	Various

3. WEB DEVELOPMENT

A. Create a Web project	Chapters 7 and 12
B. Create resources in appropriate J2EE locations	Chapters 7–14
C. Understand classpath and module dependencies	Chapter 16
D. Use Page Designer to add and modify HTML and JSP content, including the use of JavaScript	Chapters 6 and 12
E. Use content assist	Chapters 6 and 12
F. Work with Web Application Deployment Descriptor using web.xml editor	Chapters 7 and 13
G. Use Web Page Wizards	Chapters 6 and 12

4. RUNNING APPLICATIONS

A. Create server instance and server configuration	Chapters 7 and 8
B. Configure data sources	Chapter 15
C. Understand session manager	Chapters 10 and 16
D. Run/operate server	Chapters 7–15
E. Export J2EE applications	Chapter 16

5. DEBUGGING

A. Set breakpoints	Chapter 5
B. Step through code	Chapter 5
C. View variables (e.g., display expressions and use inspectors)	Chapter 5
D. Create and run code in Scrapbook Page	Chapter 5
E. Perform JSP debugging	Chapter 12

6. DATA

A. Create connection and load catalog into a project	Chapter 15
B. Create schema and generate DDL	Chapter 15

Summary

Throughout the book there are many examples to take you through the aspects of Web application development with WebSphere Studio. Most of the chapters end with a significant exercise that walks you through the steps of performing the key tasks covered in those chapters. We are confident that the total package will significantly improve your understanding and competency at developing, testing, debugging, and deploying Web components using WebSphere Studio.