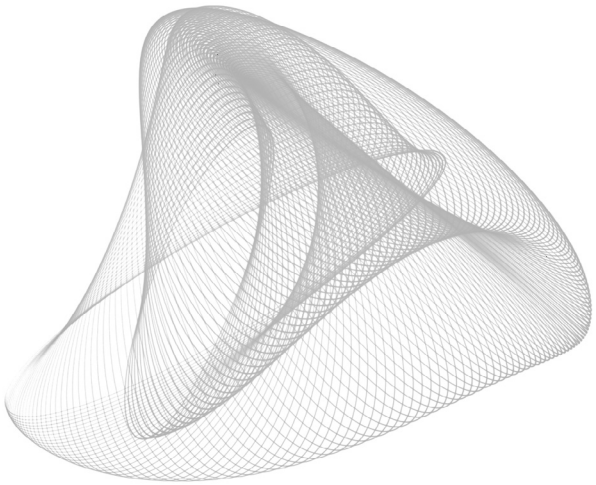
IBM® WebSphere® Application Server Programming



About the Author

Bassem (Max) Jamaleddine is a Web Systems Engineer with five years of diversified contractual experience with IBM. He administered and orchestrated the development of several main projects at IBM's T.J.Watson Research Center, from the Java-based network computer to the new generation of IBM's WAS technology.

A strong mathematical background has helped foster his belief that the soul of understanding new computer technology lies in disciplined coding coupled with proper system administration.

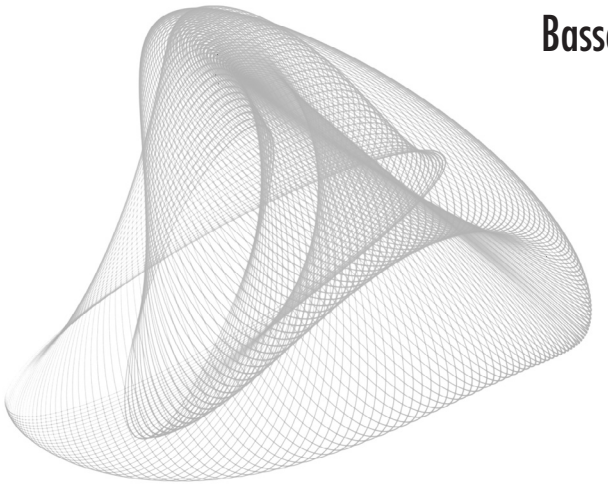
His UNIX knowledge dates from the early stages of PDP-11 system programming to the IBM SuperPower Parallel machines geared by newer AIX technologies. He has lectured at the City University of New York on advanced system programming, and on compiler and assembler construction. During the dot.com explosion, he was a major Object Oriented developer for Wall Street e-businesses, building user registration, authentication, shopping carts, and tracking and debugging routines. His forte is the development of algorithms for pattern recognition and calligraphy, such as the grammar to recognize chromosomes and digital signatures.

He earned an M.S. in Computer Science from the City University of New York, a B.S. in Computer Mathematics from Lebanese American University, and a mathematics diploma from International College in Lebanon.

He started development of Project-9 at Total Computing & Network Design, Inc. He is actively consulting on the East Coast for WebSphere deployment.

IBM® WebSphere® Application Server Programming

Bassem W. Jamaleddine



McGraw-Hill/Osborne

New York Chicago San Francisco
Lisbon London Madrid Mexico City Milan
New Delhi San Juan Seoul Singapore Sydney Toronto

The McGraw-Hill Companies

McGraw-Hill/Osborne
2600 Tenth Street
Berkeley, California 94710
U.S.A.

To arrange bulk purchase discounts for sales promotions, premiums, or fund-raisers, please contact **McGraw-Hill**/Osborne at the above address. For information on translations or book distributors outside the U.S.A., please see the International Contact Information page immediately following the index of this book.

IBM® WebSphere® Application Server Programming

1234567890 CUS CUS 0198765432

ISBN 0-07-222459-2

Publisher	Brandon A. Nordin
Vice President & Associate Publisher	Scott Rogers
Editorial Director	Wendy Rinaldi
Project Editor	Elizabeth Seymour
Technical Editor	Chris Moss
Copy Editor	Chrisa Hotchkiss
Proofreader	Paul Medoff
Indexer	Valerie Perry
Computer Designers	Carie Abrew, Lucie Ericksen, Apollo Publishing Services
Illustrators	Michael Mueller, Lyssa Wald
Series Designer	Roberta Steele
Cover Series Designer	Greg Scott
Cover Illustration	Akira Inoue/Photonica

This book was composed with Corel VENTURA™ Publisher.

This book is dedicated in loving memory to my sister,
Dima Jamaleddine

Contents at a Glance

Contents

Acknowledgments

My thanks are extended to all those who stood by me during the writing and editing of this work; to those who exercised patience at my obstinacy in re-working the text for greater clarity, and to those who, sometimes, lost patience with me. I am particularly grateful to Chris Moss who, all the way from Down Under, carefully checked my ramblings for clarity and correctness, and even found WASMON a ripper. I am deeply indebted to Elizabeth Seymour and Chrisa Hotchkiss, as well, for having glanced fair enough at each and every letter in this manuscript.

Foreword

This text explains the fundamentals of the functionality and programming of WebSphere Application Server (WAS) that are essential for readers whose aim is programming or administering the product. The text is intended to give the reader an understanding of the basic functions of WAS from the computer system perspective. The explanation follows a practical approach to understanding WAS independently of a specific version. Therefore, the text shows a (programming) methodology that is applicable to WAS v3.5, v4, and to the upcoming release of WAS v5.

As necessary background, the reader should possess programming ability in some high-level language. One full course in Java programming and some rudimentary knowledge of Perl and shell script programming constitute the level of competence required for all parts of this text. Programming for WAS is not confined solely to Java. Because J2EE application servers and J2EE application programming demand the editing of many metafiles and descriptor files, we have chosen to use Perl, shell, the UNIX make utility, and other UNIX tools, to do the text preprocessing, compilation, and deployment of code. The choice of UNIX as a development environment is justified since it is the only system that promotes application development in enterprises.

Basic application server programming is taught in the context of distributed systems that vary in platforms; therefore, various operating systems were considered. Although the Java machine is common to all platforms, threads management is operating system dependent and greatly impacts the performance of the application server. Here we have chosen AIX and Linux as platforms to highlight this difference.

Explanation of the structure of WebSphere Application Server has been made generic for the most part, without targeting any specific version of the product. The text addresses the fundamentals of WAS programming and integration, and offers many utility scripts that can be used regardless of the specific version of the product. WAS AE stands for WAS Advanced Edition. The version implied by the term WAS AE refers to either v3.5 or v4. WAS AEs stands for Advanced Edition Single Server v4. The code and utility scripts represented herein are backward compatible with WAS v5.

The reader will use WAS AEs in most of the programming (Part III), because this version is the most adequate (and is free of charge) for writing J2EE applications based on the IBM J2EE application server.

J2EE is not treated as a separate topic; rather, it is introduced in the context of web application programming and deployment under WAS. This is done very specifically, based on the assumption that most readers will probably receive a complete treatment of J2EE in another course or on their own by consulting online documentation.

The reader who would like to write web application programs and test them with WAS may begin with Part III of this book. The only requirement is a Linux sandbox with a connection to the Internet to download the WAS AEs trial version, the IBM UDB v7.2 trial version, and the code distribution of this text. No other third party software is required.

Part III can be used on its own, provided a system administrator has prepared the development environment to be used by the reader, as explained in chapters 3 and 4, or alternatively, as it is explained in Appendix A.

Material has been sequenced scrupulously to avoid forward references and to present sufficient steps that allow the reader to program with WAS AEs' free trial (but fully functional) version on a home computer. The approach taken here is to present the reader with some mechanisms to build web applications using IBM's J2EE application server—WebSphere Application Server—without the interference of any other WebSphere product. In this way, the reader will gain a clear understanding of WAS, making the IBM application server an openly available product that can, for example, be taught in a college course.

The text also breaks the dependency on using a graphical interface for programming, debugging, deployment, and administration. This is achieved by applying Perl scripts, and by using applications such as Lynx and the WebSphere Control Language (known as WSCP, and replaced by wsadmin in WAS v5). In addition, many other standard UNIX commands are used, which should be at the fingertips of any competent programmer. Using Lynx is appropriate because it provides an instantaneous view to the HTML header. Perl is an easy language that permits the automation of text processing, and therefore fits to generate the deployment descriptor of J2EE applications automatically.

The material has been classified into six parts so the text can be used in a university for two full courses as an introduction to WAS. Part I targets the preparation of the development environment; Part II supplies the requisite knowledge to administer WAS; Part III teaches programming for WAS; Part IV teaches stress-testing and debugging; Part V examines WAS monitoring, tuning, and risk management; finally, Part VI forms the appendixes.

The manuscript was written at Total Computing & Network Design, Inc. (TCND), in New York City, and was carefully reviewed by several IT professionals at IBM. TCND provided the computing facilities on which programming and testing took place. It also provided the tools and applications that helped generate the code listings.

The Gramercy Toolkit and the WASLED/WASMON application are the result of Project-9, likewise conducted at TCND. The toolkit and the application are freely distributed with the purchase of this text, and licensing is waived solely for educational use (non-commercial use): that is, specifically to accredited Computer Science departments and/or readers who will use the product on their home computers.

The most recent update of Gramercy Toolkit and WASLED/WASMON may be obtained from the website www.tcnd.com. The author welcomes readers' comments and critiques about the project, and asks that they be sent to wasbook@tcnd.com.

Bassem W. Jamaleddine
Total Computing & Network Design, Inc.
New York, New York
November 2002