

## **Rational Manual Tester User Guide**

Rational Application Developer for WebSphere Software V8 Programming Guide  
Computers, Control & Information Theory  
Practice in Software Adaption and Maintenance  
Rational Application Developer V7 Programming Guide  
The Rational Guide to Microsoft Office Access 2007 Templates  
Building SOA Solutions Using the Rational SDP  
Course Notes  
Succeeding with Use Cases  
Principles and Guidelines in Software User Interface Design  
Building J2EE Applications with the Rational Unified Process  
Visual Modeling with Rational Rose 2000 and UML  
Software Test Engineering with IBM Rational Functional Tester  
Object-oriented Methods  
Doing Hard Time  
IBM Midmarket Software Buying and Selling Guide  
Model-based Testing Vs. Capture Coverage Analysis  
Just Enough Software Test Automation  
Economica  
Aspect-oriented Software Development with Use Cases  
The Rational Unified Process  
Rational Manual Therapies  
Energy Research Abstracts  
Designing Software Product Lines with UML  
Modeling XML Applications with UML  
Rational Characterization of Pavement Structures Using Deflection Analysis: Research results & findings  
Applying Use Case Driven Object Modeling with UML  
Real-time Design Patterns  
Collaborative Application Lifecycle Management with IBM Rational Products  
Object Modeling and User Interface Design  
Software Configuration Management Strategies and Rational ClearCase  
ACM SIGGRAPH '89 Course Notes  
ATL Internals  
Smaart V8 User Guide  
Clinician's Guide to Psychological Assessment and Testing  
LIMDEP User's Manual and Reference Guide  
Refactoring  
MDA Distilled  
The Rational Unified Process Made Easy  
TestNG Beginner's Guide  
A Comprehensive Guide to Enterprise Mobility

## **Rational Application Developer for WebSphere Software V8 Programming Guide**

### **Computers, Control & Information Theory**

"[The author] explores all aspects of user interface design including the high-level conceptual model, dialog styles, organization of functionality to support user tasks, and effective error handling. She offers guidance for the design of traditional and state-of-the-art user interfaces. Both theoretical foundation and guidelines are covered in this easy reference, for ready assistance in making design decisions on the job and for deeper understanding of the topic."--Back cover.

### **Practice in Software Adaption and Maintenance**

## **Rational Application Developer V7 Programming Guide**

bull; Reflects all of the changes that were integrated into RUP v2003-the latest version of the very popular product bull; Learn the key concepts, fundamentals of structure, integral content, and motivation behind the RUP bull; Covers all phases of the software development lifecycle -from concept, to delivery, to revision

## **The Rational Guide to Microsoft Office Access 2007 Templates**

From best-selling author and noted teacher and speaker Yehuda Berg comes a thought-provoking call to action on our current global crisis. Positing that our collective abdication of responsibility — in every facet of our lives, including business and the economy, the environment, government and politics, healthcare, education, and religion — has contributed to the problems and challenges we face, Berg asserts that taking responsibility for our actions (or lack thereof) and their consequences is the key to achieving change for the better. Berg urges readers to access the power within each of us, using the principles of Kabbalah, in order to create the consciousness shift required for lasting positive change.

## **Building SOA Solutions Using the Rational SDP**

In this book, Microsoft Program Manager Zac Woodall shows developers and IT pros how to use the new template architecture in Access 2007 to deliver rich data-centric tracking applications. He explains what Access templates are, how they are used, and how to create them. His discussion of template design and customization includes details that every template designer should know, such as design guidelines, best practices, and advice for overcoming technical hurdles. Advanced topics include personalization, designing for SharePoint, template security, enterprise deployment, and a detailed breakdown of the template (.accdt) file format. In only 256 pages, this book helps the reader to: Design Access 2007 Templates Build Enterprise Templates for Access 2007 Implement Secure Access 2007 Templates This book comes with a bonus chapter "Customizing the Field Templates Task Pane" and five bonus sample templates, which readers can use in their own projects (Access Add-In, Access Ribbon XML, Contacts Modified, Table Creator, and Template Maker). These valuable bonus materials are available as a web site download.

## **Course Notes**

As the application of object technology--particularly the Java programming language--has become commonplace, a new problem has emerged to confront the software development community. Significant numbers of poorly designed programs have been created by less-experienced developers, resulting in applications that are inefficient and hard to maintain and

extend. Increasingly, software system professionals are discovering just how difficult it is to work with these inherited, "non-optimal" applications. For several years, expert-level object programmers have employed a growing collection of techniques to improve the structural integrity and performance of such existing software programs. Referred to as "refactoring," these practices have remained in the domain of experts because no attempt has been made to transcribe the lore into a form that all developers could use. . .until now. In *Refactoring: Improving the Design of Existing Code*, renowned object technology mentor Martin Fowler breaks new ground, demystifying these master practices and demonstrating how software practitioners can realize the significant benefits of this new process. With proper training a skilled system designer can take a bad design and rework it into well-designed, robust code. In this book, Martin Fowler shows you where opportunities for refactoring typically can be found, and how to go about reworking a bad design into a good one. Each refactoring step is simple--seemingly too simple to be worth doing. Refactoring may involve moving a field from one class to another, or pulling some code out of a method to turn it into its own method, or even pushing some code up or down a hierarchy. While these individual steps may seem elementary, the cumulative effect of such small changes can radically improve the design. Refactoring is a proven way to prevent software decay. In addition to discussing the various techniques of refactoring, the author provides a detailed catalog of more than seventy proven refactorings with helpful pointers that teach you when to apply them; step-by-step instructions for applying each refactoring; and an example illustrating how the refactoring works. The illustrative examples are written in Java, but the ideas are applicable to any object-oriented programming language.

## **Succeeding with Use Cases**

### **Principles and Guidelines in Software User Interface Design**

"A refreshingly new approach toward improving use-case modeling by fortifying it with aspect orientation." --Ramnivas Laddad, author of *AspectJ in Action* "Since the 1980s, use cases have been a way to bring users into software design, but translating use cases into software has been an art, at best, because user goods often don't respect code boundaries. Now that aspect-oriented programming (AOP) can express crosscutting concerns directly in code, the man who developed use cases has proposed step-by-step methods for recognizing crosscutting concerns in use cases and writing the code in separate modules. If these methods are at all fruitful in your design and development practice, they will make a big difference in software quality for developers and users alike. --Wes Isberg, AspectJ team member" "This book not only provides ideas and examples of what aspect-oriented software development is but how it can be utilized in a real development project." --Michael Ward, ThoughtWorks, Inc. "No system has ever been designed from scratch perfectly; every system is composed of features layered in top of features that accumulate over time. Conventional design techniques do not handle this well, and over time the integrity of most systems degrades as a result. For the first time, here is a set of

techniques that facilitates composition of behavior that not only allows systems to be defined in terms of layered functionality but composition is at the very heart of the approach. This book is an important advance in modern methodology and is certain to influence the direction of software engineering in the next decade, just as Object-Oriented Software Engineering influenced the last." --Kurt Bittner, IBM Corporation"Use cases are an excellent means to capture system requirements and drive a user-centric view of system development and testing. This book offers a comprehensive guide on explicit use-case-driven development from early requirements modeling to design and implementation. It provides a simple yet rich set of guidelines to realize use-case models using aspect-oriented design and programming. It is a valuable resource to researchers and practitioners alike." --Dr. Awais Rashid, Lancaster University, U.K., and author of Aspect-Oriented Database Systems "AOSD is important technology that will help developers produce better systems. Unfortunately, it has not been obvious how to integrate AOSD across a project's lifecycle. This book shatters that barrier, providing concrete examples on how to use AOSD from requirements analysis through testing." --Charles B. Haley, research fellow, The Open University, U.K. Aspect-oriented programming (AOP) is a revolutionary new way to think about software engineering. AOP was introduced to address crosscutting concerns such as security, logging, persistence, debugging, tracing, distribution, performance monitoring, and exception handling in a more effective manner. Unlike conventional development techniques, which scatter the implementation of each concern into multiple classes, aspect-oriented programming localizes them. Aspect-oriented software development (AOSD) uses this approach to create a better modularity for functional and nonfunctional requirements, platform specifics, and more, allowing you to build more understandable systems that are easier to configure and extend to meet the evolving needs of stakeholders. In this highly anticipated new book, Ivar Jacobson and Pan-Wei Ng demonstrate how to apply use cases--a mature and systematic approach to focusing on stakeholder concerns--and aspect-orientation in building robust and extensible systems. Throughout the book, the authors employ a single, real-world example of a hotel management information system to make the described theories and practices concrete and understandable. The authors show how to identify, design, implement, test, and refactor use-case modules, as well as extend them. They also demonstrate how to design use-case modules with the Unified Modeling Language (UML)--emphasizing enhancements made in UML 2.0--and how to achieve use-case modularity using aspect technologies, notably AspectJ. Key topics include Making the case for use cases and aspects Capturing and modeling concerns with use cases Keeping concerns separate with use-case modules Modeling use-cases slices and aspects using the newest extensions to the UML notation Applying use cases and aspects in projects Whatever your level of experience with aspect-oriented programming, Aspect-Oriented Software Development with Use Cases will teach you how to develop better software by embracing the paradigm shift to AOSD.

### **Building J2EE Applications with the Rational Unified Process**

Early to market, this will be one of the first books to cover ActiveX controls and ATL and will be the first classroom-tested

book on this technology. The ActiveX Template Library (ATL) is a class library for Visual C++: pre-built code modules that speed development. Using a functional approach, this book addresses the real-world choices facing Windows developers and provides practical solutions.

### **Visual Modeling with Rational Rose 2000 and UML**

Doing Hard Time is written to facilitate the daunting process of developing real-time systems. It presents an embedded systems programming methodology that has been proven successful in practice. The process outlined in this book allows application developers to apply practical techniques - garnered from the mainstream areas of object-oriented software development - to meet the demanding qualifications of real-time programming. Bruce Douglass offers ideas that are up-to-date with the latest concepts and trends in programming. By using the industry standard Unified Modeling Language (UML), as well as the best practices from object technology, he guides you through the intricacies and specifics of real-time systems development. Important topics such as schedulability, behavioral patterns, and real-time frameworks are demystified, empowering you to become a more effective real-time programmer.

### **Software Test Engineering with IBM Rational Functional Tester**

### **Object-oriented Methods**

Since the previous edition of this popular and comprehensive book was published, there have been massive changes in the field of object technology. This book has been fully revised and updated to reflect the newest technologies and methodologies, including extensive coverage of middleware, components, Java & UML. If you are a developer or manager needing to succeed with objects, this book will give you a full understanding of the key concepts, benefits and pitfalls - plus what technologies and tools are available and how to evaluate them. It offers invaluable insights into the philosophy and real-world practice of today's leading object-oriented techniques and products. Major features of this edition: detailed chapter covering middleware and migration strategies chapter describing best practice for analysis and design, with in-depth focus on architecture and patterns plus a concise presentation of the Catalysis method for component based development revised coverage of requirements, featuring detailed description of the SOMA approach coverage of Java, in addition to other object-oriented programming languages Plus:- significantly revised coverage of object-oriented databases to address new and increasingly mature products- review of processes and project management including RUP and OPEN Process, and guidance on testing and UI design- new appendices summarizing the UML notation and background survey of 50 object oriented methods- self-test questions and model answers on accompanying web-site: [www.trireme.com](http://www.trireme.com)

## Doing Hard Time

PLEASE PROVIDE SUMMARY

## IBM Midmarket Software Buying and Selling Guide

IBM® Rational® Application Developer for WebSphere® Software V8 is the full-function Eclipse 3.6 technology-based development platform for developing Java™ Platform, Standard Edition Version 6 (Java SE 6) and Java Platform, Enterprise Edition Version 6 (Java EE 6) applications. Beyond this function, Rational Application Developer provides development tools for technologies, such as OSGi, Service Component Architecture (SCA), Web 2.0, and XML. It has a focus on applications to be deployed to IBM WebSphere Application Server and IBM WebSphere Portal. Rational Application Developer provides integrated development tools for all development roles, including web developers, Java developers, business analysts, architects, and enterprise programmers. This IBM Redbooks® publication is a programming guide that highlights the features and tooling included with Rational Application Developer V8.0.1. Many of the chapters provide working examples that demonstrate how to use the tooling to develop applications and achieve the benefits of visual and rapid application development. This publication is an update of Rational Application Developer V7.5 Programming Guide, SG24-7672.

## Model-based Testing Vs. Capture Coverage Analysis

IBM® Rational® Application Developer for WebSphere® Software V7.0 (for short, Rational Application Developer) is the full function Eclipse 3.2 based development platform for developing Java™ 2 Platform Standard Edition (J2SETM ) and Java 2 Platform Enterprise Edition (J2EETM ) applications with a focus on applications to be deployed to IBM WebSphere Application Server and IBM WebSphere Portal. Rational Application Developer provides integrated development tools for all development roles, including Web developers, Java developers, business analysts, architects, and enterprise programmers. Rational Application Developer is part of the IBM Rational Software Delivery Platform (SDP), which contains products in four life cycle categories: - Architecture management, which includes integrated development environments (Application Developer is here) - Change and release management - Process and portfolio management - Quality management This IBM Redbooks® publication is a programming guide that highlights the features and tooling included with Rational Application Developer V7.0. Many of the chapters provide working examples that demonstrate how to use the tooling to develop applications, as well as achieve the benefits of visual and rapid application development. This publication is an update of Rational Application Developer V6 Programming Guide, SG24-6449. This book consists of six parts: - Introduction to Rational Application Developer - Develop applications - Test and debug applications - Deploy and profile applications - Team development - Appendixes

## **Just Enough Software Test Automation**

### **Economica**

The authors explain the underlying software development principles behind theRUP, and guide readers in its application in their organization.

## **Aspect-oriented Software Development with Use Cases**

This revised and enlarged edition of a classic in Old Testament scholarship reflects the most up-to-date research on the prophetic books and offers substantially expanded discussions of important new insight on Isaiah and the other prophets.

## **The Rational Unified Process**

MDA Distilled is an accessible introduction to the MDA standard and its tools and technologies. The book describes the fundamental features of MDA, how they fit together, and how you can use them in your organization today. You will also learn how to define a model-driven process for a project involving multiple platforms, implement that process, and then test the resulting system.

## **Rational Manual Therapies**

The IBM® Midmarket Software Buying and Selling Guide is tailored specifically to help the management and IT staff of small and mid-sized businesses evaluate how the IBM midmarket portfolio can provide simple and cost-effective solutions to common business problems. Along with a midmarket customer focus, this IBM Redpaper™ publication is designed to help IBM teams and Business Partners be more effective in serving small and mid-sized businesses. We illustrate how IBM software for the midmarket can help businesses use the Web to reduce expenses, improve customer service, and expand into new markets. We cover the IBM software offering for the midmarket, which includes what the software does, the platforms it runs on, where to find more information, and how it can help your business become more profitable: - IBM Business Partners often keep a printed copy of this guide in their briefcases for software references - Customers can view this guide online and look up software-value messages and IBM product family offering comparisons - IBM Sales Representatives can print parts of this guide as "leave-behinds" for customers, to give them extra collateral on midmarket software of interest To make sure that you have the latest version of this guide, download it from this web address:

<http://www.redbooks.ibm.com/abstracts/redp3975.html?Open>

## **Energy Research Abstracts**

### **Designing Software Product Lines with UML**

In this study, a rational overlay design method for flexible pavements in Arizona has been developed which includes roughness, fatigue and plastic deformation models. The method is incorporated in a microcomputer program which is also capable of analyzing the economics of other rehabilitation alternatives. During the development of the method, twenty in-service pavement sites were selected from Arizona highways covering various geographical and environmental regions, soil types, pavement conditions and traffic volumes. nondestructive tests (NDT) were performed on these sites using the Falling Weight Deflectometer at three stress levels as well as the Dynaflect. The pavement layers and subgrade moduli were backcalculated from NDT data using both static and dynamic analyses. The moduli of asphaltic layers were further adjusted for temperature. Statistical analysis was performed and the stress sensitivity was found to be small. Cone Penetration tests were further performed to verify the subgrade moduli. Asphalt concrete cores, base and subbase samples and undisturbed subgrade materials. In addition, base and subbase gradation as well as soil classification and Atterberg limits were obtained. The study expanded the data base for material properties in Arizona. This volume is the first in a three volume set. Volume II provides field testing procedure and workstation development and Volume III is a computer user guide.

### **Modeling XML Applications with UML**

"This is the fourth report on mothers and babies in NSW to combine the annual reports of the NSW Midwives Data Collection (MDC), the Neonatal Intensive Care Units' Data Collection and the NSW Birth Defects Register." -- p.9.

### **Rational Characterization of Pavement Structures Using Deflection Analysis: Research results & findings**

Offers advice on designing and implementing a software test automation infrastructure, and identifies what current popular testing approaches can and cannot accomplish. Rejecting the automation life cycle model, the authors favor limited automation of unit, integration, and system testing. They also present a control synchronized data-driven framework to help jump-start an automation project. Examples are provided in the Rational suite test studio, and source code is available at a supporting web site. Annotation copyrighted by Book News, Inc., Portland, OR.

## Applying Use Case Driven Object Modeling with UML

An introduction to the core practices of software configuration management, a software engineering discipline that secures against cost over runs while insuring project timeliness and quality.

## Real-time Design Patterns

"Designing Software Product Lines with UML is well-written, informative, and addresses a very important topic. It is a valuable contribution to the literature in this area, and offers practical guidance for software architects and engineers."  
--Alan Brown Distinguished Engineer, Rational Software, IBM Software Group "Gomaa's process and UML extensions allow development teams to focus on feature-oriented development and provide a basis for improving the level of reuse across multiple software development efforts. This book will be valuable to any software development professional who needs to manage across projects and wants to focus on creating software that is consistent, reusable, and modular in nature."  
--Jeffrey S Hammond Group Marketing Manager, Rational Software, IBM Software Group "This book brings together a good range of concepts for understanding software product lines and provides an organized method for developing product lines using object-oriented techniques with the UML. Once again, Hassan has done an excellent job in balancing the needs of both experienced and novice software engineers." --Robert G. Pettit IV, Ph.D. Adjunct Professor of Software Engineering, George Mason University "This breakthrough book provides a comprehensive step-by-step approach on how to develop software product lines, which is of great strategic benefit to industry. The development of software product lines enables significant reuse of software architectures. Practitioners will benefit from the well-defined PLUS process and rich case studies." --Hurley V. Blankenship II Program Manager, Justice and Public Safety, Science Applications International Corporation "The Product Line UML based Software engineering (PLUS) is leading edge. With the author's wide experience and deep knowledge, PLUS is well harmonized with architectural and design pattern technologies." --Michael Shin Assistant Professor, Texas Tech University Long a standard practice in traditional manufacturing, the concept of product lines is quickly earning recognition in the software industry. A software product line is a family of systems that shares a common set of core technical assets with preplanned extensions and variations to address the needs of specific customers or market segments. When skillfully implemented, a product line strategy can yield enormous gains in productivity, quality, and time-to-market. Studies indicate that if three or more systems with a degree of common functionality are to be developed, a product-line approach is significantly more cost-effective. To model and design families of systems, the analysis and design concepts for single product systems need to be extended to support product lines. Designing Software Product Lines with UML shows how to employ the latest version of the industry-standard Unified Modeling Language (UML 2.0) to reuse software requirements and architectures rather than starting the development of each new system from scratch. Through real-world case studies, the book illustrates the fundamental concepts and technologies used in the design and

implementation of software product lines. This book describes a new UML-based software design method for product lines called PLUS (Product Line UML-based Software engineering). PLUS provides a set of concepts and techniques to extend UML-based design methods and processes for single systems in a new dimension to address software product lines. Using PLUS, the objective is to explicitly model the commonality and variability in a software product line. Hassan Gomaa explores how each of the UML modeling views--use case, static, state machine, and interaction modeling--can be extended to address software product families. He also discusses how software architectural patterns can be used to develop a reusable component-based architecture for a product line and how to express this architecture as a UML platform-independent model that can then be mapped to a platform-specific model. Key topics include: Software product line engineering process, which extends the Unified Development Software Process to address software product lines Use case modeling, including modeling the common and variable functionality of a product line Incorporating feature modeling into UML for modeling common, optional, and alternative product line features Static modeling, including modeling the boundary of the product line and information-intensive entity classes Dynamic modeling, including using interaction modeling to address use-case variability State machines for modeling state-dependent variability Modeling class variability using inheritance and parameterization Software architectural patterns for product lines Component-based distributed design using the new UML 2.0 capability for modeling components, connectors, ports, and provided and required interfaces Detailed case studies giving a step-by-step solution to real-world product line problems Designing Software Product Lines with UML is an invaluable resource for all designers and developers in this growing field. The information, technology, and case studies presented here show how to harness the promise of software product lines and the practicality of the UML to take software design, quality, and efficiency to the next level. An enhanced online index allows readers to quickly and easily search the entire text for specific topics.

### **Collaborative Application Lifecycle Management with IBM Rational Products**

From Rational Acoustics, the owners & developers Smaart, comes the official Smaart v8 User Guide. The Smaart v8 User Guide is a comprehensive guide to working with the latest version of professional audio's most widely used system analysis & optimization software. All of Smaart v8's measurement capabilities are covered in detail, along with helpful illustrations and application examples. It also includes sections on fundamental audio concepts, navigating the user interface, capturing & managing data as well as an extensive set of appendices covering measurement rig setup, licensing & installation, applicable standards and suggested further reading. Written in Rational Acoustics' approachable easy-to-read style, with just the right amount of geeky humor, the Smaart v8 User Guide is more than just a software manual, it is a fantastic all-in-one reference that Smaart users will find themselves returning to again and again.

### **Object Modeling and User Interface Design**

"Overall, this is an excellent guide to the use and administration of psychological tests. It provides straightforward directions and instructions on how to utilize testing in such a way as to better inform clinical practice. I could see this book as a mainstay on any counselor's bookshelf, especially those who are seeking a way to utilize standardized testing in their practice."--The Professional Counselor Journal

Finally, a detailed and crystal clear guide to psychological assessment that effectively integrates 'best practices' with the realities of negotiating the mental health care system and insurance providers. I plan to draw on this practical guide in my private practice and to incorporate it as a required text in my advanced counseling assessment classes at both the master's and doctoral level. This book is a treasure for any mental health professional involved in psychological assessment.

Joseph G. Ponterotto, PhD Professor of Counseling Psychology, Fordham University

Standardized psychological testing is often essential for reliably determining the presence of a wide range of psychiatric and personality disorders, along with effectively addressing related issues that may require a psychological referral. This nuts-and-bolts guide to conducting efficient and accurate psychological testing in clinical settings provides mental health professionals with experienced guidance in the entire process, and includes a complete set of forms and templates for all aspects of assessment and testing, from the initial referral and diagnostic interview to the final report. Based on the author's experience with over two thousand psychological and neuropsychological testing cases, this highly practical book presents a standardized process of assessment, testing, interpretation, report-writing, and presenting feedback to patients, family members, and other professionals. Actual case examples of patients from a wide age range illustrate the assessment and testing process in action. The text provides printed and electronic versions of referral and related forms, initial psychological assessment report templates that include critical areas of coverage for obtaining insurance approval, and interpretation tables for an exceptional inventory of key standardized psychological tests. Integral to the book is a review of psychological tests in seven key categories that most effectively address differential diagnostic dilemmas and related referral questions that clinicians are likely to encounter in practice. It also provides effective strategies for selecting the appropriate tests based on the particular diagnostic questions, guidance for successfully obtaining insurance approval for a targeted yet feasible number of testing hours, and an efficient system for simultaneous test interpretation and report writing.

**Key Features:** Includes an overview of the assessment process, from the initial referral to completion of the final report  
Features effective reviews of commonly used tests, including neuropsychological, intelligence, personality, and behavioral inventories  
Includes print and digital templates and forms for all phases of assessment and testing  
Aids clinicians in both private practice and other health care settings to work within managed care and be effectively reimbursed for services  
Includes information on conducting forensic competency to stand trial assessments, including the author's new measure of assessing a defendant's understanding of the legal system

### **Software Configuration Management Strategies and Rational ClearCase**

"Object Modeling and User Interface Design merges theories with practical techniques to create methods for the design to

today's systems. By reading this book you will gain an understanding of the benefits of integrating object-oriented analysis approaches with human computer interaction design, and learn how to systematically design interactive systems for their human users."--BOOK JACKET.

### **ACM SIGGRAPH '89 Course Notes**

Praise for Software Test Engineering with IBM Rational Functional Tester The Indispensable Resource for Automated Testing Automated software testing has become a critical exercise, especially for developers utilizing iterative and agile methods. However, to achieve the full benefits of automated testing, teams need a deep understanding of both its principles and their testing tools. If you're among the thousands of developers using IBM Rational Functional Tester (RFT), this book brings together all the insight, examples, and real-world solutions you need to succeed. Eight leading IBM testing experts thoroughly introduce this state-of-the-art product, covering issues ranging from building test environments through executing the most complex and powerful tests. Drawing on decades of experience with IBM Rational testing products, they address both technical and nontechnical challenges and present everything from best practices to reusable code. Coverage Includes Integrating IBM RFT into your development processes Building highly efficient test environments, test harnesses, and test scripts Using RFT Visual Editor to extend testing automation to novice users Mastering basic scripting techniques, from data capture to script synchronization Managing script data using RFT Datapools Efficiently debugging scripts using Eclipse™ or Visual Studio® Managing execution flow: playback settings, logic, error handling, and more Handling domains that are not supported by RFT Using advanced techniques, such as mouse delays and custom verification points Testing specialized software, including mainframe, SAP, Siebel, and Adobe® Flex® applications Extending RFT with external libraries Developing RFT support for third-party Java™ or .NET controls Using RFT in both Linux® and Windows® environments Configuring internationalized testing within the RFT framework

### **ATL Internals**

This book is written in a friendly, beginner's guide style with plenty of step-by-step instructions with appropriate examples. This book is great for developers and testers who are new to TestNg and want to learn how to use TestNG for writing their application as well as functional tests. This book assumes that you have experience in Java and OOPs concepts and have worked with certain IDE.

### **Smaart V8 User Guide**

## **Clinician's Guide to Psychological Assessment and Testing**

XML is rapidly becoming the standard platform for delivering e-Business information and integrating e-Business systems. XML developers desperately need mature software development processes and tools for developing effective applications. David Carlson fills the gap, showing exactly how to leverage the worldwide UML standard for modeling complex systems in advanced XML development. In Modeling XML Applications with UML, he presents the first comprehensive framework for modeling communications in any B2B software system. Carlson presents in-depth coverage of UML-based analysis, design, and modeling of XML content within e-Business environments. The book includes detailed coverage of using UML to support the creation of new XML-based B2B vocabularies and industry portals that reflect the requirements of several key stakeholder communities, including consumers, business analysts, web application specialists, system integration specialists, and content developers. Carlson presents several B2B use cases, and then decomposes them into scenarios illustrated with class diagrams, sequence diagrams, and activity diagrams showing how XML fits into an overall e-Business solution. Each chapter concludes with "steps for success" that distill UML's general principles into specific recommendations for action.

## **LIMDEP User's Manual and Reference Guide**

### **Refactoring**

### **MDA Distilled**

## **The Rational Unified Process Made Easy**

The first UML book to cover Rational Rose 2000, this brand-new edition reviews the three key interrelated components of state-of-the-art software system design: the Rational Unified process, the Unified Modeling Language, and Rational Rose 2000. Then, through a simplified case study, it walks developers through a real-world business system. Includes screen shots demonstrating UML at work in the Rational Rose 2000 modeling tool.

## **TestNG Beginner's Guide**

Although enterprise mobility is in high demand across domains, an absence of experts who have worked on enterprise mobility has resulted in a lack of books on the subject. A Comprehensive Guide to Enterprise Mobility fills this void. It supplies authoritative guidance on all aspects of enterprise mobility-from technical aspects and applications to

## **A Comprehensive Guide to Enterprise Mobility**

[ROMANCE](#) [ACTION & ADVENTURE](#) [MYSTERY & THRILLER](#) [BIOGRAPHIES & HISTORY](#) [CHILDREN'S](#) [YOUNG ADULT](#) [FANTASY](#)  
[HISTORICAL FICTION](#) [HORROR](#) [LITERARY FICTION](#) [NON-FICTION](#) [SCIENCE FICTION](#)