Praise for the First Edition

For anyone considering Groovy, or just interested in seeing what all of the fuss is around the features of dynamic languages, this book will deliver.

—Gregory Pierce, JavaLobby.org

Not just a language guide, this book presents the clear, readable, and enjoyable specification of Groovy … you should definitely read it.

—Alexander Popescu, Mindstorm

A real page-turner. Brilliant examples … all other programming books I know really fall behind.

—Dr. Gernot Starke

Excellent code samples … very readable.

—Scott Shaw, ThoughtWorks

Great, logical focus on language features.

—Norman Richards, author of *XDoclet in Action*

Destined to be the definitive guide. First rate!

—Glen Smith, Bytecode Pty Ltd.

Examples are clear, complete, and they work!

—David Sills, JavaLobby.org

Among the top five Manning books. For me personally, it's also a perception-changing and influential book.

—Weiqi Gao

The examples are the strongest part of the book—all assumptions are checked using assertions, and they have been run before printing so one can trust that they're faultless. Explanations are fine-grained so even inexperienced developers can read it with understanding.

—Marek Zganiacz, Comarch SA

Very readable, engaging, and does a great job of slotting Groovy into the broader world of software development. Highly recommended.

—Pan Pantziarka

Real computer LITERATURE.

—Johannes Link

To our families

Groovy in Action

Second Edition

DIERK KÖNIG
PAUL KING

WITH
GUILLAUME LAFORGE
HAMLET D'ARCY
CÉDRIC CHAMPEAU
ERIK PRAGT
AND JON SKEET

MANNING
SHELTER ISLAND

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Development editor: Nermina Miller
Copyeditor: Jodie Allen
Technical editor Michael Smolyak
Proofreader: Elizabeth Martin
Technical proofreader: Gordon Dickens
Typesetter: Dennis Dalinnik
Cover designer: Marija Tudor

ISBN: 9781935182443
Printed in the United States of America
1 2 3 4 5 6 7 8 9 10 – EBM – 20 19 18 17 16 15

brief contents

contents

foreword to the first edition

I first integrated Groovy into a project I was working on almost two years ago. There is a long and rich history of using "scripting languages" as a flexible glue to stitch together, in different ways, large modular components from a variety of frameworks. Groovy is a particularly interesting language from this tradition, because it doesn't shy away from linguistic sophistication in the pursuit of concise programming, especially in the areas around XML, where it is particularly strong. Groovy goes beyond the "glue" tradition of the scripting world to being an effective implementation language in its own right. In fact, while Groovy is often thought of and referred to as a scripting language, it really is much more than that.

It is traditional for scripting languages to have an uneasy relationship with the underlying linguistic system in which the frameworks are implemented. In Groovy's case, they have been able to leverage the underlying Java model to get integration that is smooth and efficient. And because of the linguistic similarities between Java and Groovy, it is fairly painless for developers to shift between programming in one environment and the other.

Groovy in Action by Dierk König and his coauthors is a clear and detailed exposition of what is groovy about Groovy. I'm glad to have it on my bookshelf.

JAMES GOSLING
CREATOR OF JAVA
DECEMBER 2006

preface

Nothing is more terrible than ignorance in action.

—Johann Wolfgang von Goethe

Thinking back to January 2007 when the first edition of this book hit the shelves, feels like time travel to the Middle Ages. The idea of using a programming language other than Java on the Java platform was widely considered frivolous. Today, a new language seems to pop up every other week, and we even go as far as designing languages for specific domains (DSLs) on a per-project basis.

This evolution of languages reflects a change in concerns. If performance were still our utmost concern, we would all be coding in a low-level language. But if performance is considered "good enough" for our purposes, we now turn our focus on human approachability.

Groovy has been a trendsetter for this development. Many Groovy features that ease the burden of developers are now commonplace in novel languages and may even find their way into newer versions of Java: literal declarations for common datatypes, simplified property access, null-safe dereferencing, closures, and more. Surprisingly many languages have adopted Groovy's optional typing strategy—few languages can claim to have static and dynamic *behavior* at the same time, though, the way Groovy has since version 2.

Just like Groovy, the first edition of this book set some trends as well. The idea of having every single listing as a self-testing piece of code resonated in the market and may be one reason why the book is among Manning's top-ten bestsellers of the decade.

The feedback for the first edition was overwhelming. We never expected to have so many great developers speaking so nicely about our work. We have no words to express this feeling of being proud and humbled at the same time. Most touching, though, was the stranger who once gave Dierk a pat on the back and mumbled, "Thank you for the book!" and then disappeared into the crowd. This book is for him.

We are fully aware that the first edition would have never been so successful if Groovy itself had been less appealing. The reason for Groovy's success is easy to see: it delivers its power in the most Java-friendly manner. It is Java's dynamic friend.

The development of Groovy, from version 1.0 covered in the first edition of this book until the current version 2.4, has closed what used to be a syntax gap by providing enums, annotations, generics, the classic `for` loop, nested classes, varargs, static imports, and the ability to use Groovy closures where Java 8 expects lambda expressions.

The Groovy project has progressed at a very high speed, not only in its core but also at its periphery. We see, for example, new usages of compile-time metaprogramming. This core feature gets instantly applied in the Spock testing framework, which in turn contributes back its "power assert" feature to the core. The community is buzzing and it has become a challenge to keep up to date with all the developments and activities.

It's only natural that many readers of the first edition of *Groovy in Action* (or "Gina" as we say for short) demanded an update that we are now happy to deliver as the second edition (codename "ReGina"). Our goal in this book is not only to rework the code examples, update the API description, and explain new features, but also to reflect the marketplace and the growth of the ecosystem. Groovy has evolved from a niche language to the default choice for dynamic programming on the Java platform for millions of developers.

Major financial organizations use Groovy to transfer billions of dollars every day, space agencies watch the stars with the help of Groovy, and satellite live-data streams are handled by Groovy code. Groovy is traveling the oceans, shipping containers around the globe, helping software developers automate recurring tasks, and running Mom's website. We felt an obligation to provide an up-to-date, solid, and comprehensive book to all these users.

Not only did Groovy and its environment change, we authors changed as well. We enjoyed the luxury of working on Groovy projects, introducing new team members to the language, running workshops and tutorials, recognizing struggles (and occasionally struggling ourselves), finding lots of unanticipated use cases while consulting, exploring new practices, using the toolset in anger, and generally facing the Groovy development reality. The book reflects these experiences.

In this second edition, we put more emphasis on the optional typing system, explain both dynamic and static metaprogramming in full depth, dive into type checking and static compilation, cover domain-specific languages, and introduce new modules that have evolved for user interfaces, testing, XML, JSON, database programing,

Web Services, dependency management, build automation, and concurrent programming as well as give you an updated overview of the Groovy ecosystem. We hope you will find this updated book an enjoyable and rewarding read.

DIERK KÖNIG
PAUL KING

acknowledgments

Our publisher warned us that a second edition would be much more difficult. We did not understand that back then, but he was right. We needed to get more coauthors on board to account for the growth of Groovy and we are very grateful that Hamlet D'Arcy, Cédric Champeau, and Erik Pragt joined the group. Paul King invested an enormous amount of extra time and I (Dierk) am also very grateful to him for that.

We're deeply indebted to our technical reviewing team: Atul Khot, David McFarland, Jakob Mayr, Ken Shih, Paul Grebenc, Phillip Warner, Rick Wagner, Robert O'Connor, Ronald Tischliar, Scott Ruch, Vinod Panicker, and Vladimír Oraný, with special thanks to our technical editor Michael Smolyak and technical proofreader Gordon Dickens.

While the book was in development, readers could subscribe to Manning's Early Access Program (MEAP) to get the content early and to provide feedback. We received so many valuable suggestions that we cannot possibly list everyone's name, but we would like to say a big thank you to all of you! The MEAP ran longer than any other and while we are not proud of that record, we thank everyone for their patience and hope that you will find the book up-to-date and worth the wait.

Other friends helped with the book in one way or another: Andres Almiray, Bob Brown, Nick Chase, Andy Clement, Scott Davis, Marc Guillemot, Dr. Urs Hengartner, Arturo Herrero, Martin Huber, Roshan Dawrani, Wim Deblauwe, Dean DeChambeau, Gordon Dickens, Andrew Eisenberg, Jeremy Flowers, Dave Klein, Rupin Kotecha, Kenneth Kousen, Peter Ledbrook, Mac Liaw, Johannes Link, Joshua Logan, Chris Mair, Tsuyoshi Miyake, Vaclav Pech, Graeme Rocher, Baruch Sadogursky, Uwe Sauerbrei,

Erik Schwalbe, Larry Seltzer, Jim Shingler, Dan Sline, Glen Smith, David Stuve, Andre Steingress, Jochen Theodorou, Marija Tudor, Craig Walls, Dr. Hans-Dirk Walter, and Geertjan Wielenga.

The book would never had made it to the shelves without the support and guidance of everyone at Manning, especially our publisher Marjan Bace, our editors Nermina Miller and Maureen Spencer, and all the other great people who worked with us: Jodie Allen, Luke Bace, Jeff Bleiel, Olivia Booth, Candace Gillhoolley, Todd Green, Steven Hong, Cynthia Kane, Emily Macel, Elizabeth Martin, Tara McGoldrick Walsh, Mary Piergies, Christina Rudloff, Mike Stephens, and Kevin Sullivan.

Finally, very special thanks to James Gosling for writing the foreword to the first edition of *Groovy in Action*.

But most of all, we thank our families for their ongoing encouragement to pursue our ideas, their patience when we were once again physically or mentally absent, and their love that gives us a purpose in life. We love you.

about this book

Groovy in Action, Second Edition describes the Groovy language, presents the library classes and methods that Groovy adds to the standard Java Development Kit, and leads you through a number of topics that you are likely to encounter in your daily development work. The book has three parts:

- Part 1 The Groovy language
- Part 2 Around the Groovy library
- Part 3 Applied Groovy

An introductory chapter explains what Groovy is and then part 1 starts with a broad overview of Groovy's language features, before going into more depth about scalar and collective datatypes. The language description includes an explanation of the closure concept that is ubiquitous in Groovy, describing how it relates to and distinguishes itself from control structures. We present Groovy's model of object-orientation and its dynamic capabilities at both runtime and compile-time. Part 1 closes with a surprise: You can use Groovy as a static language as well!

Part 2 begins the library description with a presentation of Groovy's builder concept and its various implementations. An explanation of the GDK follows, with Groovy's enhancements to the Java standard library. This is the "beef" of the library description in part 2. The Groovy library shines with simple but powerful support for database programming and XML and JSON handling, and we include a detailed exposition of both topics. Another big advantage of Groovy is its all-out seamless

integration with Java, and we explain the options provided by the Groovy library for setting this into action.

If part 1 was a tutorial and part 2 a reference, part 3 is about typical use cases for Groovy. It starts with a thorough exposition of how to use Groovy for test automation. Testing is an important topic in itself, but with Groovy even more so since Groovy developers seem to be very quality-oriented and even in otherwise plain-Java projects, Groovy is often used for testing because it is so convenient. Next, we want to use Groovy on multi-core machines and thus go into concurrent programming with Groovy. Another much-requested topic is using Groovy for domain specific languages, which we cover in a full, dedicated chapter. Part 3 ends with an overview of the Groovy ecosystem.

The book closes with an extensive series of helpful appendixes, which are intended to serve as quick references, cheat sheets, and detailed technical descriptions.

Who should read this book?

This book is for everyone who wants to learn Groovy as a new dynamic programming language. Existing Groovy users can use it to deepen their knowledge; and both new and experienced programmers can use it as a black-and-white reference. We found ourselves going to our own book to look up details that we had forgotten. Newcomers to Groovy will need a basic understanding of Java since Groovy is completely dependent on it; Java basics are not covered in our book.

Topics have been included that will make reading and understanding easier, but are not mandatory prerequisites: patterns of object-oriented design, Ant, Maven, JUnit, HTML, XML, JSON, Swing, and JavaFX. It is beneficial—but not required—to have been exposed to some other scripting language. This enables you to connect what you read to what you already know. Where appropriate, we point out similarities and differences between Groovy and other languages.

What's new in the second edition?

When starting the second edition, we considered adding visual clues or icons to the book so readers could quickly see what had changed from the first edition. We had to give up on that idea or the whole book would have been full of markers since there is hardly any paragraph that hasn't changed!

The second edition is a full rewrite. We dropped some chapters, reorganized others, and added new ones, so the book now has 20 chapters, up from 16, and a few hundred additional pages of genuinely new content. These changes reflect the evolution of the language and its use in the market.

Tackling the task of covering such a big topic needs many hands and we were very lucky that Hamlet Darcy, Cédric Champeau, and Erik Pragt joined the team. Hamlet authored the new chapters 9 "AST Transformations" and 20 "The Groovy Ecosystem." Cédric contributed his deep knowledge of Groovy internals to the new chapter 10 "Groovy as a static language" and helped to fine-tune chapters 7, 9, and 16. Erik got

the laborious task of going through all changes to the Groovy standard library for chapter 12 "Working with the GDK" and fundamentally revised chapter 17 "Unit testing with Groovy" to cover the popular Spock testing framework.

Guillaume Laforge revised chapter 16 "Integrating Groovy" and shaped new chapter 19 "Domain Specific Languages (DSLs)" to address this important usage of Groovy.

Dierk König added chapter 19 "Concurrent Groovy with GPars" to show how well Groovy fits into the multi-core era. He also thoroughly revised and updated the core "language" chapters 1 through 6. Former chapter 7 was split into "Object orientation, Groovy style," and a new chapter 8 "Dynamic Programming with Groovy."

Paul King revised the "library" chapters 11 "Working with builders," 13 "Database programming with Groovy," and split the former XML chapter 14 into "Working with XML and JSON" and 15 "Interacting with Web Services" and extended the content to account for the rising importance of these Groovy usages. He also did the enormous work of going through every single page of the book to ensure consistency in style, wording, feel, and appearance. With so many authors and such diverse topics it is very difficult to keep the book coherent. If we finally managed to achieve this, it is thanks to Paul.

Code conventions and downloads

This book provides copious examples that show how you can make use of each of the topics covered. Source code in listings or in text appears in a `fixed-width font like this` to separate it from ordinary text. In addition, class and method names, object properties, and other code-related terms and content in text are presented using `fixed-width font`.

Occasionally, code is italicized, as in *`reference`*`.dump()`. In this case *`reference`* should not be entered literally but replaced with the content that is required, such as the appropriate reference.

Where the text contains the pronouns "I" and "we", the "we" refers to all the authors. "I" refers to the lead author of the respective chapter.

Most of the code examples contain Groovy code. This code is very compact so we present it "as is" without any omissions. Unless stated otherwise, you can copy and paste it into a new file and run it right away. In rare cases, when this wasn't possible, we have used … (ellipses).

Java, HTML, XML, and command-line input can be verbose. In many cases, the original source code (available online) has been reformatted; we've added line breaks and reworked indentation to accommodate the page space available in the book. In rare cases, when even this was not enough, line-continuation markers were added.

Code annotations accompany many of the listings, highlighting important concepts. In some cases, numbered cueballs link to additional explanations that follow the listing.

You can download the source code for all of the examples in the book from the publisher's website at www.manning.com/GroovyinActionSecondEdition.

Keeping up to date

The world doesn't stop turning when you finish writing a book, and getting the book through production also takes time. Therefore, some of the information in any technical book becomes quickly outdated, especially in the dynamic world of agile languages.

This book covers Groovy 2.4. Groovy will see numerous improvements, and by the time you read this, it's possible that an updated version will have become available. New Groovy versions always come with a detailed list of changes. It is unlikely that any of the core Groovy concepts as laid out in this book will change significantly in the near future; and even then the emphasis is likely to be on *additional* concepts and features. Groovy has earned a reputation of caring deeply about backward compatibility. This outlook makes the book a wise investment, even in a rapidly changing world.

We will do our best to keep the online resources for this book reasonably up to date and provide information about language and library changes as the project moves on. Please check for updates on the book's web page at www.manning.com/GroovyinActionSecondEdition.

Author Online

Purchase of *Groovy in Action* includes free access to a private web forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the authors and from other users. To access the forum and subscribe to it, point your web browser to www.manning.com/GroovyinActionSecondEdition. This page provides information on how to get on the forum once you are registered, what kind of help is available, and the rules of conduct on the forum. It also provides links to the source code for the examples in the book, errata, and other downloads.

Manning's commitment to our readers is to provide a venue where a meaningful dialog between individual readers and between readers and the authors can take place. It is not a commitment to any specific amount of participation on the part of the authors, whose contribution to the AO remains voluntary (and unpaid). We suggest you try asking the authors some challenging questions lest their interest stray!

The Author Online forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

About the cover illustration

The figure on the cover of *Groovy in Action, Second Edition* is a "Danzerina del Japon," a Japanese dancer, taken from a Spanish compendium of regional dress customs first published in Madrid in 1799. While the artist may have captured the "spirit" of a Japanese dancer in his drawing, the illustration does not accurately portray the looks, dress, or comportment of a Japanese woman or geisha of the time, compared to Japanese drawings from the same period. The artwork in this collection was clearly not researched first hand!

The book's title page states:

> *Coleccion general de los Trages que usan actualmente todas las Nacionas del Mundo desubierto, dibujados y grabados con la mayor exactitud por R.M.V.A.R. Obra muy util y en special para los que tienen la del viajero universal*

which we translate, as literally as possible, thus:

> *General collection of costumes currently used in the nations of the known world, designed and printed with great exactitude by R.M.V.A.R. This work is very useful especially for those who hold themselves to be universal travelers*

Although nothing is known of the designers, engravers, and workers who colored this illustration by hand, the "exactitude" of their execution is evident in this drawing. The "Danzerina del Japon" is just one of many figures in this colorful collection. Travel for pleasure was a relatively new phenomenon at the time and books such as this one were popular, introducing both the tourist as well as the armchair traveler to the exotic inhabitants, real and imagined, of other regions of the world.

Dress codes have changed since then and the diversity by nation and by region, so rich at the time, has faded away. It is now often hard to tell the inhabitant of one continent from another. Perhaps, trying to view it optimistically, we have traded a cultural and visual diversity for a more varied personal life. Or a more varied and interesting intellectual and technical life.

We at Manning celebrate the inventiveness, the initiative, and the fun of the computer business with book covers based on the rich diversity of regional life two centuries ago, brought back to life by the pictures from this collection.

about the authors

DIERK KÖNIG has worked for over 20 years as a professional software developer, architect, trainer, and consultant. Through his publications, conference appearances, trainings, workshops, and consulting activities, Dierk has reached more developers than he ever thought possible. He has worked with Canoo Engineering AG, Basle, Switzerland, since 2000, where he is a cofounder and enjoys being part of a thriving organization.

Dierk contributes to many open source projects, including Groovy, Grails, OpenDolphin, Frege, and CanooWebTest. He joined the Groovy project in 2004 and has worked as a committer ever since. He presented Groovy to win the JAX Innovation Award 2007 and won the JAX Developer Challenge 2009 with his team.

He is an acknowledged reviewer and contributor to numerous books, including the classic *Extreme Programming Explained* (Kent Beck), *Test-Driven Development* (Kent Beck), *Agile Development in the Large* (Jutta Eckstein), *Unit Testing in Java* (Johannes Link), *JUnit and Fit* (Frank Westphal), *Refactoring in Large Software Projects* (Martin Lippert and Stephen Roock), *The Definitive Guide to Grails* (Graeme Rocher), and *Grails in Action* (Glen Smith, Peter Ledbrook).

In the course of authoring this second edition, Dierk became a happy husband and a proud father of a girl and a boy. You can follow him on twitter as @mittie.

DR. PAUL KING'S career spans technical and managerial roles in a number of organizations, underpinned by deep knowledge of the information technology and telecommunications markets and a passion for the creation of innovative organizations.

Throughout his career, Paul has provided technical and strategic consulting to hundreds of organizations in the U.S. and Asia Pacific. The early stages of Paul's career were highlighted by his contributions to various research fields, including object-oriented software development, formal methods, telecommunications, and distributed systems. He has had numerous publications at international conferences and in journals and trade magazines. He is an award-winning author and sought-after speaker at conferences.

Currently, Paul leads ASERT (Advanced Software Engineering, Research & Training), which is recognized as a world-class center of expertise in the areas of middleware technology, agile development, and internet application development and deployment. ASERT has experience teaching thousands of students in more than 15 countries, and has provided consulting services and development assistance throughout Asia Pacific to high-profile startups and government e-commerce sites. In his spare time, Paul is a taxi driver and homework assistant for his seven children and two grandchildren. You can follow him on twitter as @paulk_asert.

GUILLAUME LAFORGE has been the official Groovy project manager since the end of 2004, after having been a contributor and later a core committer on the project. He is also the specification lead for JSR-241, the ongoing effort to standardize the Groovy language through Sun's Java Community Process. Guillaume is Groovy's "public face" and often responds to interviews regarding Groovy and presents his project at conferences around the world, such as at JavaOne or Devoxx, where he recently spoke about how scripting can simplify enterprise development. Guillaume cofounded the G2One company, which focused on and further developed the Groovy and Grails technologies, later acquired by SpringSource; also VMware and its Pivotal spin-off. Guillaume recently joined Restlet as Product Ninja and Advocate.

CÉDRIC CHAMPEAU is a member of the Groovy core team. He is a passionate developer who started writing programs at the age of eight and learned it the hard way by manually typing magazine listings into an Amstrad PC1512. He worked several years in natural language processing where he used Groovy in multiple contexts, from a workflow engine to a DSL for linguists, and Lucene custom scoring. This is how he dived into the internals of the language and started contributing before becoming one of the core team members. He implemented many advanced Groovy features like compilation customizers, static compilation, traits, the markup template engine, and the recent support for Android.

HAMLET D'ARCY is a software engineer at Microsoft, founder of the Basel-based Hackergarten open source coding group, and can be found speaking at local and international user groups and conferences. He's a committer on the Groovy and CodeNarc projects and a contributor on a number of other projects (including the IDEA Groovy Plugin). He's passionate about learning new languages and different ways of thinking about problems. He blogs regularly at http://hamletdarcy.blogspot.com.

ERIK PRAGT is a passionate software developer with a broad range of experience in static languages like Java and Scala, and dynamic languages like Groovy, JavaScript, and Python. Having worked as a consultant for a broad range of customers, mostly in the Telecom, ISP, and banking sectors, Erik is now an independent freelance consultant. He founded the Dutch Groovy and Grails user group, and is a regular conference speaker and trainer. Erik spends most of his free time working on open source software. In the limited time he's not sitting behind the computer he can be found in the gym, riding his motorcycle, or diving, always looking for new inspiration, which he shares on twitter at @epragt.

JON SKEET Jon Skeet is a software engineer working for Google in London. He is probably best known for his contributions on Stack Overflow. He blogs, tweets (@jonskeet), speaks at conferences, and generally says too much and listens too little. For some years now, his primary open source contribution to the world has been Noda Time, a better .NET date and time API. He is the author of Manning's *C# in Depth, Third Edition*.