

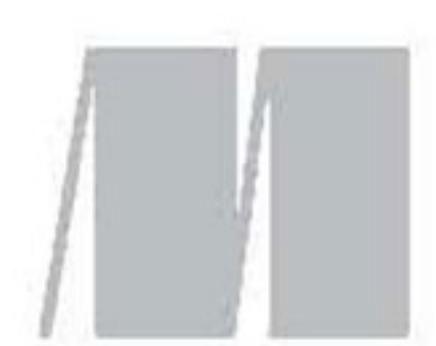
Covers iText 5

A painting of a woman in traditional Spanish dress, known as a mantilla, standing on a red and white striped floor. She is wearing a dark blue velvet dress with a gold-trimmed lace panel at the waist and a gold skirt decorated with small blue beads. She holds a small blue book in her hands.

# iText IN ACTION

SECOND EDITION

Bruno Lowagie

 MANNING



# *contents*

---

*preface* xvii  
*preface to the first edition* xviv  
*acknowledgments* xxi  
*about this book* xxiii  
*about the title* xxviii  
*about the cover illustration* xxix

## PART 1 CREATING PDF DOCUMENTS FROM SCRATCH.....1

### **1** *Introducing PDF and iText* 3

- 1.1 Things you can do with PDF 4
- 1.2 Working with the examples in this book 6
- 1.3 Creating a PDF document in five steps with iText 8

*Creating a new Document object* 9 ▪ *Getting a PdfWriter instance* 12 ▪ *Opening the Document* 13 ▪ *Adding content* 15 ▪ *Closing the Document* 17

- 1.4 Summary 19

## 2 Using iText's basic building blocks 20

- 2.1 Illustrating the examples with a real-world database 21
- 2.2 Adding Chunk, Phrase, Paragraph, and List objects 22
  - The Chunk object: a String, a Font, and some attributes* 23 ▪ *The Phrase object: a List of Chunks with leading* 25 ▪ *Paragraph object: a Phrase with extra properties and a newline* 30 ▪ *Distributing text over different lines* 34 ▪ *The List object: a sequence of Paragraphs called ListItem* 37
  - The DrawInterface: vertical position marks, separators, and tabs* 41
- 2.3 Adding Anchor, Image, Chapter, and Section objects 46
  - The Anchor object: internal and external links* 47 ▪ *Chapter and Section: get bookmarks for free* 49 ▪ *The Image object: adding raster format illustrations* 51 ▪ *Summary* 56

## 3 Adding content at absolute positions 57

- 3.1 Introducing the concept of direct content 58
  - Direct content layers* 58 ▪ *Graphics state and text state* 60 ▪ *A real-world database: three more tables* 62
- 3.2 Adding text at absolute positions 67
  - Convenience method: PdfContentByte.showTextAligned()* 67
  - Convenience method: ColumnText.showTextAligned()* 70
- 3.3 Working with the ColumnText object 74
  - Using ColumnText in text mode* 75 ▪ *Using ColumnText in composite mode* 81
- 3.4 Creating reusable content 84
  - Image XObjects* 84 ▪ *The PdfTemplate object* 87
- 3.5 Summary 92

## 4 Organizing content in tables 93

- 4.1 Constructing tables 94
  - Your first PdfPTable* 94 ▪ *PdfPTable properties* 95
- 4.2 Changing the properties of a cell 98
  - PdfPCell in text mode* 99 ▪ *PdfPCell in composite mode* 107
- 4.3 Dealing with large tables 112
  - Repeating headers and footers* 112 ▪ *Splitting tables* 113
  - Memory management for LargeElement implementations* 115
- 4.4 Adding a table at an absolute position 116
  - Working with writeSelectedRows()* 116 ▪ *Wrapping tables in columns* 119
- 4.5 Summary 121

## 5 *Table, cell, and page events* 122

- 5.1 Decorating tables using table and cell events 123
  - Implementing the PdfPTableEvent interface* 123 ▪ *Implementing the PdfPCellEvent interface* 125 ▪ *Combining table and cell events* 128
- 5.2 Events for basic building blocks 133
  - Generic Chunk functionality* 134 ▪ *Paragraph events* 138 ▪ *Chapter and Section events* 138 ▪ *Page order and blank pages* 141
- 5.3 Overview of the page boundaries 142
  - The mediabox* 143 ▪ *The cropbox* 145 ▪ *Otherpage boundaries* 147
- 5.4 Adding page events to PdfWriter 148
  - Adding a header and a footer* 148 ▪ *Solving the “page X of Y” problem* 150 ▪ *Adding a watermark* 152 ▪ *Creating a slideshow* 154
- 5.5 Summary 155

## PART 2 MANIPULATING EXISTING PDF DOCUMENTS ... 157

## 6 *Working with existing PDFs* 159

- 6.1 Accessing an existing PDF with PdfReader 160
  - Retrieving information about the document and its pages* 160
  - Reducing the memory use of PdfReader* 163
- 6.2 Copying pages from existing PDF documents 164
  - Importing pages* 165 ▪ *Scaling and superimposing pages* 167
  - N-up copying and tiling PDF documents* 170
- 6.3 Adding content with PdfStamper 173
  - Adding content at absolute positions* 173 ▪ *Creating a PDF in multiple passes* 175 ▪ *Adding company stationery to an existing document* 176
  - Inserting pages into an existing document* 178 ▪ *Filling out a PDF form* 179
- 6.4 Copying pages with PdfCopy 186
  - Concatenating and splitting PDF documents* 187 ▪ *PdfCopy versus PdfSmartCopy* 189 ▪ *Concatenating forms* 191
- 6.5 Summary 192

## 7 *Making documents interactive* 194

- 7.1 Introducing actions 195
  - Document-navigation actions* 195 ▪ *Explicit destinations* 199
  - JavaScript in PDF documents* 202 ▪ *More actions* 205

## *Part 1*

# *Creating PDF documents from scratch*

---

**P**art 1 shows you how to create a document from scratch. Concepts such as iText's basic building blocks and direct content will be introduced, and important objects for adding columns and tables to a document are discussed in great detail. These first five chapters also explain how to add finishing touches to your document, using page events for headers, footers, page numbers, and watermarks.



# *Introducing PDF and iText*

---



## **This chapter covers**

- A summary of what will be presented in this book
- Compiling and executing your first example
- Learning the five steps in iText's PDF creation process

Call me Bruno. About ten years ago—never mind how long precisely—I thought I'd create a small PDF library in Java and publish it as free and open source software (F/OSS). Little did I know that this would lead to my writing a whale of a book about the extensive functionality that has been added over the years.

That library was iText, and the book was titled *iText in Action: Creating and Manipulating PDF* (2007). Today, iText is the world's leading F/OSS PDF library. It's released under the Affero General Public License (AGPL) and is available in two versions: the original Java version, and the C# port, iTextSharp. These libraries make it possible for you to enhance applications with dynamic PDF solutions. You can use iText to create invoices for your customers if you have a web shop, to produce tickets if you work for an airline or railway company, and so on. You can integrate iText into an application to generate PDF documents as an alternative to

printing on paper, to add digital signatures to a document, to split or concatenate different documents, and so forth.

In the first edition of *iText in Action*, readers learned why things work the way they do in iText, complemented with simple examples. This second edition takes you further with more real-life examples, skipping a bit on the whys, but presenting comprehensive code samples that you can use to solve everyday problems.

In this chapter, I'll give you a quick overview of the things you can do with PDF—you'll compile and execute a first “Hello World” example—and you'll learn the basics of creating PDFs with iText.

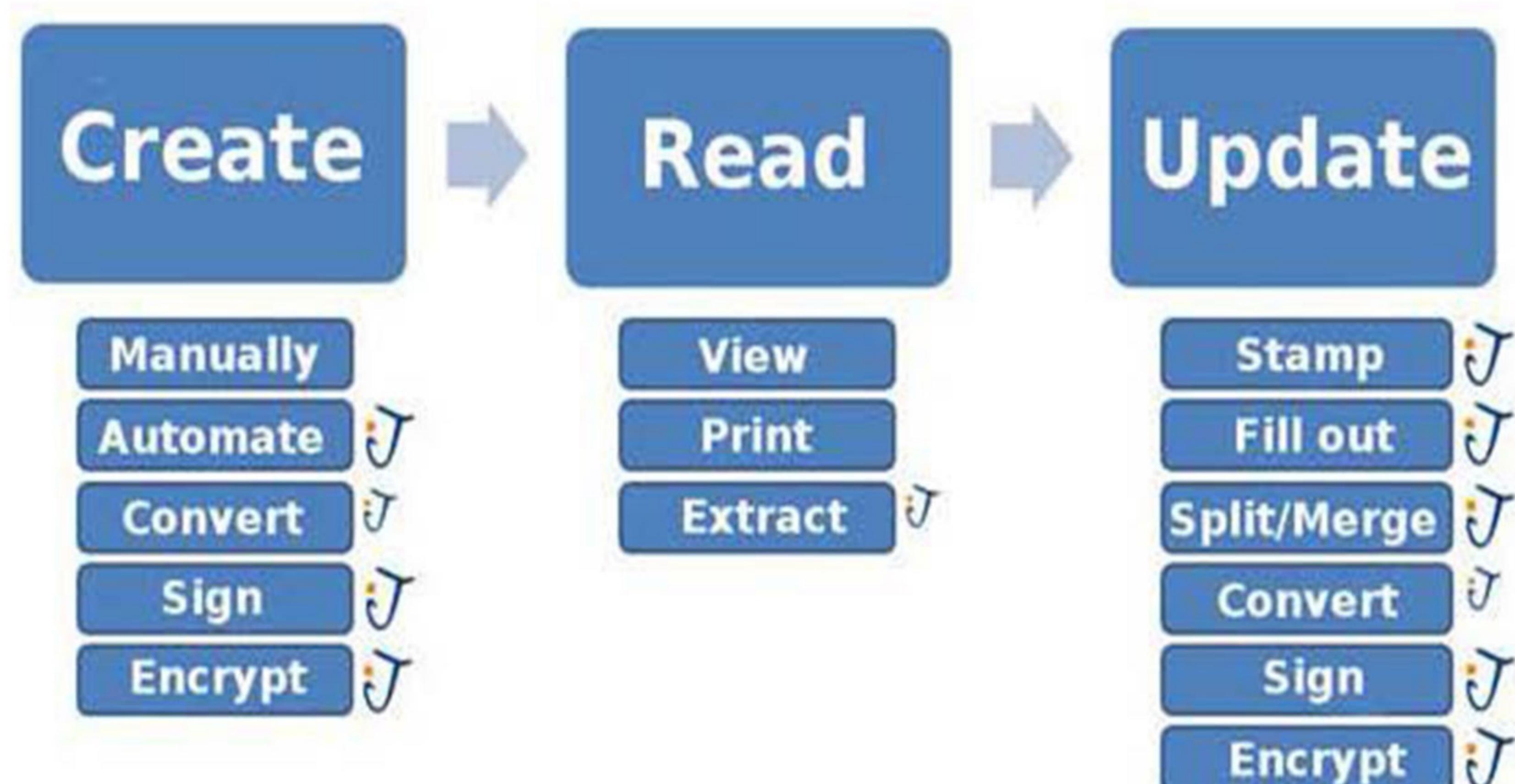
## 1.1 **Things you can do with PDF**

Let's start with six quick facts about PDF:

- PDF is the Portable Document Format.
- It's an open file format (ISO-32000-1), originally created by Adobe.
- It's used for documents that are independent of system software and hardware.
- PDF documents are an essential part of the web.
- Adobe Reader is the most widely used PDF viewer.
- There are a lot of free and proprietary, open and closed source, desktop and web-based software products for creating, viewing, and manipulating PDF documents.

Figure 1.1 offers an overview of the things you can do with PDF. There are tools to *create* PDF documents, there are applications to *consume* PDF documents, and there are utilities to *manipulate* existing PDF documents.

If you look at PDF creation, you'll find that graphical designers use desktop applications such as Adobe Acrobat or Adobe InDesign to create a document in a manual or semimanual process. In another context, PDF documents are created programmatically, using an API to produce PDFs directly from software applications, without—or with minimal—human intervention. Sometimes the document is created in an intermediary format first, then converted to PDF. These different approaches demand different software products. The same goes for PDF manipulation. You can update a PDF manually in Adobe Acrobat, but there are also tools that allow forms to be filled out automatically based on information from a database.



**Figure 1.1** Overview of PDF-related functionality. The functionality covered by iText is marked with the iText logo.

This book will focus on the automation side of things: we'll create and manipulate PDF documents in an automated process using iText. The functionality covered by iText in figure 1.1 is marked with the iText logo. A smaller logo indicates that the functionality is only partly supported.

Typically, iText is used in projects that have one of the following requirements:

- The content isn't available in advance: it's calculated based on user input or real-time database information.
- The PDF files can't be produced manually due to the massive volume of content: a large number of pages or documents.
- Documents need to be created in unattended mode, in a batch process.
- The content needs to be customized or personalized; for instance, the name of the end user has to be stamped on a number of pages.

Often you'll encounter these requirements in web applications, where content needs to be served dynamically to a browser. Normally, you'd serve this information in the form of HTML, but for some documents, PDF is preferred over HTML for better printing quality, for identical presentation on a variety of platforms, for security reasons, or to reduce the file size. In this case, you can serve PDF on the fly.

As you read this book, you'll create and manipulate hundreds of PDF documents that demonstrate how to use a specific feature, how to solve common and less common issues, and how to build an application that involves PDF technology. We'll use iText because it's an API that was developed to allow developers to do the following (and much more):

- Generate documents and reports based on data from an XML file or a database
- Create maps and books, exploiting numerous interactive features available in PDF
- Add bookmarks, page numbers, watermarks, and other features to existing PDF documents
- Split or concatenate pages from existing PDF files
- Fill out interactive forms
- Serve dynamically generated or manipulated PDF documents to a web browser

For first-time users, this book is indispensable. Although the basic functionality of iText is easy to grasp, the first parts of this book significantly lower the learning curve and gradually offer more advanced functionality.

It's also a must-have for the many developers who are already familiar with iText. In the final chapters, many PDF secrets hidden in ISO-32000-1, the open standard that defines the Portable Document Format, will be unveiled. Even experienced iText developers will learn new ways to master the PDF specification using their favorite PDF library.

Without further ado, let's start with a simple example that explains how to compile and run the many examples that come with this book.

ook  
ERT

of both iText  
ispensable.”  
mpet, Inc.

ised with  
everyone

prehensive.”  
ga  
olutions

nplex...  
is here.”

Solutions

source on

International AG

8-1-935182-61-0  
935182-61-7



2610

# iText IN ACTION Second Edition

Bruno Lowagie

**S**earch for “Java PDF” and what do you think you’ll find in the #1 position? Why, iText, of course. Ever since its launch in 2000, this open source Java library has been the most popular and most broadly used tool for programmatic creation and manipulation of PDF. With it you too can easily transform static PDF into live, interactive applications.

**iText in Action, Second Edition** is an entirely revised new version of the popular first edition. It introduces the latest version of iText, and it lowers the learning curve to its advanced features. Following its innovative, practical examples, you’ll master new form types, including AcroForm, explore the XML Forms Architecture (XFA), and discover techniques for linking documents, creating a PDF based on records in a database, and much more.

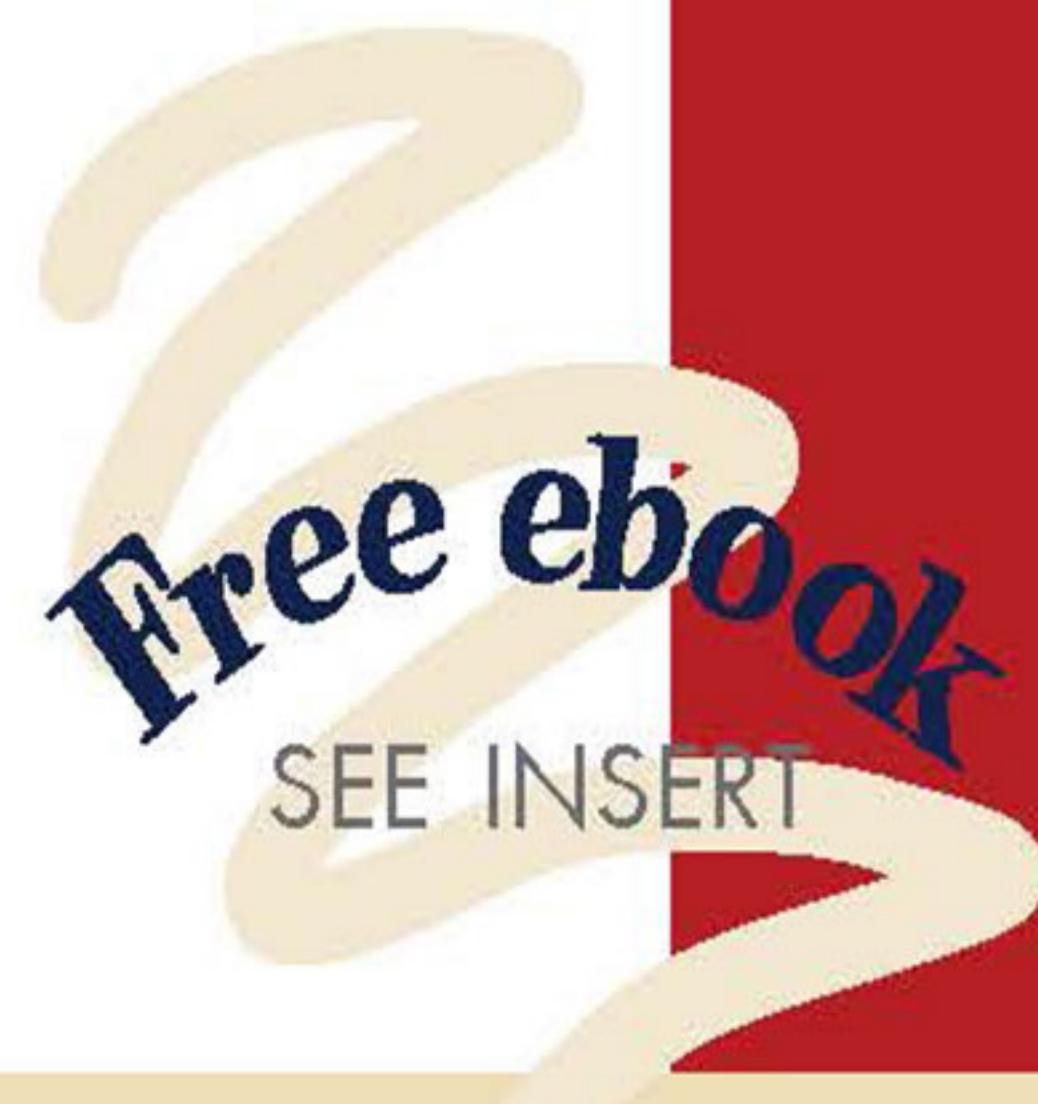
## What's Inside

- Automate static and dynamic XFA forms
- How to generate dynamic PDF from XML or a database
- How to add digital signatures
- Covers iText 5

Written by the creator of iText, this new edition covers the latest version of iText and Java 5. The examples can be easily adapted to .NET using iTextSharp or iText.NET.

**Bruno Lowagie** is the original developer and current maintainer of iText.

For a free ebook for owners of this book, go to  
[manning.com/iTextinActionSecondEdition](http://manning.com/iTextinActionSecondEdition)



“Deep coverage of both iText and PDF—indispensable.”

—Kevin Day, Trumpet, Inc.

“The classic, revised with practical code everyone can use.”

—John S. Griffin  
Overstock.com

“Masterful, comprehensive.”

—Saicharan Manga  
Services and Solutions

“Invaluable examples... what you need is here.”

—Paulo Soares  
Glintt Business Solutions

“The canonical source on iText.”

—Michael Klink  
AuthentiDate International AG



MANNING

\$59.99 / Can \$68.99 [INCLUDING eBOOK]

ISBN 13: 978-1-935182-61-0  
ISBN 10: 1-935182-61-7



9 781935 182610

