

THE EXPERT'S VOICE® IN JAVA

# Pro Java 7 NIO.2

Anghel Leonard

Apress®

# Pro Java 7 NIO.2

Anghel Leonard

Apress®

## Pro Java 7 NIO.2

Copyright © 2011 by Anghel Leonard

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN-13 (pbk): 978-1-4302-4011-2

ISBN-13 (electronic): 978-1-4302-4012-9

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning

Lead Editor: Matthew Moodie

Technical Reviewer: Michael Turner

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Coordinating Editor: Corbin Collins

Copy Editor: Bill McManus, Damon Larson

Compositor: Apress Production (Christine Ricketts)

Indexer: SPI Global

Artist: SPI Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com).

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com](http://www.apress.com).

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at [www.apress.com/bulk-sales](http://www.apress.com/bulk-sales).

Any source code or other supplementary materials referenced by the author in this text is available to readers at [www.apress.com](http://www.apress.com). For detailed information about how to locate your book's source code, go to [www.apress.com/source-code/](http://www.apress.com/source-code/).

# Contents at a Glance

<b>Contents at a Glance .....</b>	<b>iii</b>
<b>Contents .....</b>	<b>iv</b>
<b>About the Author .....</b>	<b>xiii</b>
<b>About the Technical Reviewer .....</b>	<b>xiv</b>
<b>Acknowledgments .....</b>	<b>xv</b>
<b>Preface .....</b>	<b>xvi</b>
<b>Chapter 1: Working with the Path Class .....</b>	<b>1</b>
<b>Chapter 2: Metadata File Attributes .....</b>	<b>11</b>
<b>Chapter 3: Manage Symbolic and Hard Links .....</b>	<b>35</b>
<b>Chapter 4: Files and Directories .....</b>	<b>43</b>
<b>Chapter 5: Recursive Operations: Walks .....</b>	<b>77</b>
<b>Chapter 6: Watch Service API .....</b>	<b>111</b>
<b>Chapter 7: Random Access Files .....</b>	<b>135</b>
<b>Chapter 8: The Sockets APIs .....</b>	<b>169</b>
<b>Chapter 9: The Asynchronous Channel API .....</b>	<b>215</b>
<b>Chapter 10: Important Things to Remember .....</b>	<b>263</b>
<b>Index .....</b>	<b>273</b>

# Contents

Contents at a Glance .....	iii
Contents .....	iv
About the Author .....	xiii
About the Technical Reviewer .....	xiv
Acknowledgments .....	xv
Preface .....	xvi
<b>Who This Book Is For .....</b>	<b>xvii</b>
<b>What This Book Covers .....</b>	<b>xvii</b>
Carry out Path class operations .....	xvii
Get/set file metadata through the new java.nio.file.attribute API .....	xvii
Manage symbolic and hard links .....	xviii
Deal with files and directories through the new java.nio.file.Files API .....	xviii
Use the FileVisitor API to develop recursive file operations .....	xviii
Explore the Watch Service API and file changed notifications .....	xviii
Use the new SeekableByteChannel API for working with random access file .....	xviii
Develop blocking/non-blocking socket-based applications .....	xviii
Use the jewel in the crown of NIO.2: the Asynchronous Channel API .....	xiv
Work with the Zip File System and write a custom file system provider .....	xiv
<b>What You Need to Use This Book .....</b>	<b>xvii</b>

Chapter 1: Working with the Path Class..... 1

**Introducing the Path Class ..... 1**

**Defining a Path ..... 2**

    Define an Absolute Path..... 2

    Define a Path Relative to the File Store Root..... 3

    Define a Path Relative to the Working Folder ..... 3

    Define a Path Using Shortcuts ..... 3

    Define a Path from a URI..... 4

    Define a Path using FileSystems.getDefault().getPath() Method ..... 4

    Get the Path of the Home Directory ..... 4

**Getting Information About a Path ..... 4**

    Get the Path File/Directory Name ..... 5

    Get the Path Root..... 5

    Get the Path Parent..... 5

    Get Path Name Elements ..... 5

    Get a Path Subpath ..... 5

**Converting a Path ..... 6**

    Convert a Path to a String..... 6

    Convert a Path to a URI ..... 6

    Convert a Relative Path to an Absolute Path..... 6

    Convert a Path to a Real Path ..... 6

    Convert a Path to a File..... 7

**Combining Two Paths ..... 7**

**Constructing a Path Between Two Locations ..... 8**

**Comparing Two Paths..... 9**

<b>Iterate over the Name Elements of a Path .....</b>	<b>10</b>
<b>Summary .....</b>	<b>10</b>
Chapter 2: Metadata File Attributes .....	11
<b>Supported Views in NIO.2 .....</b>	<b>11</b>
<b>Determining Views Supported by a Particular File System .....</b>	<b>12</b>
<b>Basic View .....</b>	<b>13</b>
Get Bulk Attributes with readAttributes() .....	13
Get a Single Attribute with getAttribute() .....	14
Update a Basic Attribute .....	15
<b>DOS View .....</b>	<b>16</b>
<b>File Owner View .....</b>	<b>18</b>
Set a File Owner Using Files.setOwner() .....	18
Set a File Owner Using FileOwnerAttributeView.setOwner() .....	18
Set a File Owner Using Files.setAttribute() .....	19
Get a File Owner Using FileOwnerAttributeView.getOwner() .....	19
Get a File Owner Using Files.getAttribute() .....	20
<b>POSIX View .....</b>	<b>21</b>
POSIX Permissions .....	22
POSIX Group Owner .....	22
<b>ACL View .....</b>	<b>23</b>
Read an ACL Using Files.getFileAttributeView() .....	23
Read an ACL Using Files.getAttribute() .....	24
Read ACL Entries .....	24
Grant a New Access in an ACL .....	26

<b>File Store Attributes</b> .....	<b>27</b>
Get Attributes of All File Stores.....	28
Get Attributes of the File Store in Which a File Resides .....	29
<b>User-Defined File Attributes View</b> .....	<b>30</b>
Check User-Defined Attributes Supportability .....	30
Operations on User-Defined Attributes .....	31
<b>Summary</b> .....	<b>34</b>
Chapter 3: Manage Symbolic and Hard Links .....	35
<b>Introducing Links</b> .....	<b>35</b>
<b>Creating Links from the Command Line</b> .....	<b>36</b>
<b>Creating a Symbolic Link</b> .....	<b>36</b>
<b>Creating a Hard Link</b> .....	<b>38</b>
<b>Checking a Symbolic Link</b> .....	<b>39</b>
<b>Locating the Target of a Link</b> .....	<b>40</b>
<b>Checking If a Link and a Target Point to the Same File</b> .....	<b>41</b>
<b>Summary</b> .....	<b>42</b>
Chapter 4: Files and Directories.....	43
<b>Checking Methods for Files and Directories</b> .....	<b>43</b>
Checking for the Existence of a File or Directory.....	43
Checking File Accessibility .....	44
Checking If Two Paths Point to the Same File .....	46
Checking the File Visibility.....	47



<b>Creating and Reading Directories</b> .....	<b>47</b>
Listing File System Root Directories.....	47
Creating a New Directory.....	48
Listing a Directory’s Content.....	49
<b>Creating, Reading, and Writing Files</b> .....	<b>53</b>
Using Standard Open Options.....	53
Creating a New File.....	54
Writing a Small File.....	54
Reading a Small File .....	56
Working with Buffered Streams.....	58
Working with Unbuffered Streams.....	59
<b>Creating Temporary Directories and Files</b> .....	<b>62</b>
Creating a Temporary Directory.....	62
Creating Temporary Files.....	66
Deleting, Copying, and Moving Directories and Files .....	70
Rename a File .....	75
<b>Summary</b> .....	<b>76</b>
Chapter 5: Recursive Operations: Walks.....	77
<b>The FileVisitor Interface</b> .....	<b>78</b>
FileVisitor.visitFile() Method.....	78
FileVisitor.preVisitDirectory() Method .....	78
FileVisitor.postVisitDirectory() Method.....	79
FileVisitor.visitFileFailed() Method .....	79
<b>The SimpleFileVisitor Class</b> .....	<b>79</b>
<b>Starting the Recursive Process</b> .....	<b>80</b>

<b>Common Walks</b> .....	<b>81</b>
Writing a File Search Application.....	81
Writing a File Delete Application.....	102
Writing a Copy Files Application .....	104
Writing a Move Files Application .....	107
<b>Summary</b> .....	<b>110</b>
Chapter 6: Watch Service API .....	111
<b>The Watch Service API Classes</b> .....	<b>112</b>
<b>Implementing a Watch Service</b> .....	<b>112</b>
Creating a WatchService .....	112
Registering Objects with the Watch Service.....	112
Waiting for the Incoming Events.....	114
Getting a Watch Key.....	114
Retrieving Pending Events for a Key.....	115
Retrieving the Event Type and Count.....	116
Retrieving the File Name Associated with an Event .....	117
Putting the Key Back in Ready State .....	117
Closing the Watch Service .....	118
Gluing It All Together .....	118
<b>Other Examples of Using a Watch Service</b> .....	<b>120</b>
Watching a Directory Tree .....	120
Watching a Video Camera.....	125
Watching a Printer Tray System .....	128
<b>Summary</b> .....	<b>134</b>
Chapter 7: Random Access Files .....	135