

Java 7 SE, Part 2 of 4: IO, New IO, and Network Protocols

page 1

Meet the expert: Brigitte Birze is a seasoned software development professional with over 25 years of experience in Corporate IT and Engineering across many verticals. Brigitte has been involved in every phase of the software development lifecycle from the perspective of several roles: individual contributor, team lead, S/W architect, system engineer, proposal author, and project manager. Her innovative software architectures have resulted in six published papers and eight patents. Brigitte's dynamic communication skills, paired with her depth and breadth of technical knowledge, give her the unique ability to make the complex understandable and to convey technical concepts to cross-functional groups, speaking at the business or technical level.

Prerequisites: This course assumes that the users have a solid understanding object-oriented principals and experience coding with Java 5 or greater. This course was filmed using Eclipse for RCP and RAP developers (Java developer edition). A general understanding of Eclipse is required but only to understand the methods shown. The theory for this course will work on any IDE that supports the Java 7 SE SDK. You should have view the course 'Java 7 SE: Enhancements and Concurrency' before viewing this course.

Runtime: 02:18:49

Course description: Java SE 7 platform gave us a brand new file based I/O architecture, NIO2. This new architecture solved most of the problems we saw in the original stream based I/O. You'll see how the NIO2 Path class is used to represent the underlying file system syntax, in a manner that is transparent to the application. Next you'll see how the NIO2 Files class works with the Path class. Then we'll cover the new NIO2 asynchronous channels, to let us monitor and control the completion of our asynchronous events. NIO2 also gives us a new asynchronousChannelGroup to group these channels, and supply them with a pool of threads for resource sharing. And we'll take a look at the new WatchService file change notification API, which lets you register to be notified when files or directories change on the file system. After our tour of NIO2, we'll move to the new networking features provided by Java SE 7. We'll experiment with the URLClassLoader, which now implements the Closable I/F and its close() method. Then we'll take a look at the new protocols we can now access. One is the Stream Control Transmission Protocol, or SCTP, and another new protocol is Sockets Direct Protocol, or SDP.

Course outline:

Java Stream Based File

- Introduction
- Java File I/O Issues
- Java SE 6 File I/O and NIO
- NIO2: Non-Blocking IO 2
- NIO Path and Path Classes
- Demo: Paths
- Demo: NIO
- Summary

NIO Files Class

- Introduction
- NIO File Class
- NIO File Operations
- Demo: Create Directories & Files
- Demo: Managing Metadata
- Windows ACL File Attributes
- Demo: ACL File Attributes
- Demo: New Directory ACL
- Summary

NIO File System Operations

- Introduction
- NIO File System Operations
- Demo: File System Operations
- Demo: DeletelfExists

- Summary

NIO Walking the Dir Tree

- Introduction
- NIO Walking the Dir Tree
- Demo: Walk Tree
- Demo: Copy Tree
- Summary

Finding Files With NIO

- Introduction
- Finding Files with NIO2
- NIO File Glob Arguments
- Demo: Pattern Matching
- Demo: Glob Patterns
- Summary

NIO Async Operations

- Introduction
- NIO Architecture
- NIO2 Asynchronous Channels
- Demo: Asynchronous Operations
- Summary

NIO Async Channel Group

- Introduction
- NIO Async Channel Group
- Demo: Async Channel Group
- Summary

NIO Change Notification

- Introduction

- NIO File Change Notification
- Summary

- Demo: WatchService
- Demo: WatchEvent
- Summary

Network Protocols

- Introduction
- Java SE 7: New Networking
- URLClassLoader.close() Method
- Demo: URLClassLoader
- Summary

SCTP

- Introduction
- SCTP
- SCTP NIO Channels
- SCTP Server Open Channel
- SCTP Server Accept Client Msg
- SCTP Server Respond to Client
- SCTP Client, Connect to Server
- SCTP Client, Send Message
- Summary

SDP

- Introduction
- SDP (Sockets Direct Protocol)
- Using SDP
- Invoking an SDP Application