Gradle In-Depth

Course Summary

Description

Join us for this three-day Gradle course to learn how to use the Gradle build system to increase your productivity tremendously. This Gradle training class first runs through some Groovy fundamentals and then moves quickly into in-depth coverage of the Gradle build system, including a module on how to work with the new Gradle-based Android build system as well as an introduction to using the Gradle Scala plugin.

You'll spend a significant portion of the class with your hands on the keyboard, learning by doing, working through lab exercises designed to build on the presented material and explore the depths of Gradle. This class is suitable for newcomers as well as people who already have some experience with Gradle. The class will be delivered by a Gradle Core Developer, which gives you access to the deepest Gradle knowledge available. No question will remain unanswered.

Objectives

By the end of this course, students will be able to:

- Learn the basics of Gradle and Groovy
- How Gradle integrates with Android
- Write your own Gradle plugins
- Integrate with Ant and Maven
- Learn about copy and archive tasks in Gradle
- Working with a powerful file system API
- Build migration best practices
- How to use the Gradle performance
- Learn how to build Java, Android, and Scala applications
- How to efficiently move away from your current build environment
- Learn how to deal with complex build requirements
- Learn how to deal with large multi module builds

Topics

- The Gradle Project
- The Groovy Language
- Gradle Introduction
- Tasks
- Logging
- Plugins
- Working with files
- Ant Integration
- Dependency management
- Extending the model
- Task inputs & outputs
- The Java plugin
- The Scala plugin
- The Android plugin
- Multiproject builds
- The build runtime

Audience

New and experienced users are encouraged to attend Gradle In Depth. This class has something for every level of Gradle user.

Prerequisites

This course assumes a good understanding of the Java language. Some code is initially easier to understand if you also have a basic understanding of the Groovy language. But due to Groovy's similarity to Java, Groovy is not a prerequisite. There will be a very short introduction to Groovy at the beginning of the course.

Duration

Three days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Gradle In-Depth

Course Outline

I. The Gradle Project
- A. About the project
- B. Documentation / Resources / Support

II. The Groovy Language
- A. Executing Groovy code
- B. Syntax basics
- C. Dynamic typing
- D. Domain Specific Languages & Gradle
- E. Groovy JDK extensions
- F. Closures

III. Gradle Introduction
- A. The Gradle philosophy
- B. Installing Gradle
- C. Gradle build scripts
- D. The build lifecycle
- E. The Gradle build daemon

IV. Tasks
- A. Declaring tasks
- B. Executing tasks
- C. Configuring tasks
- D. Implementing custom tasks
- E. Extending tasks
- F. Task dependencies
- G. Task exclusion
- H. Conditionally skipping tasks
- I. The Task Graph API
- J. Task rules

V. Logging
- A. Controlling log levels
- B. Logging from the build script
- C. Logging from classes
- D. Managing child process output

VI. Plugins
- A. What are Gradle plugins?
- B. The role of plugins
- C. Packaging and distributing plugins
- D. Review of core plugins

VII. Working with files
- A. The file tasks (e.g. Copy, Sync)
- B. Filtering, excluding, aggregating, renaming
- C. Working with CopySpecs
- D. Expanding and making archives (zip, tar etc.)
- E. The Gradle file APIs
- F. Best practices for working with files

VIII. Ant Integration
- A. Gradle compared to Ant
- B. When to utilize Ant in Gradle
- C. Using Ant tasks in Gradle builds
- D. Importing Ant builds
- E. Migrating from Ant to Gradle

IX. Dependency management
- A. Terminology Repositories, metadata and transitive
- B. dependencies
- C. Dependency configurations
- D. Declaring dependencies
- E. Customizing dependency resolution & conflict management
- F. Publishing artifacts (Maven & Ivy)

X. Extending the model
- A. Extending existing objects
- B. Object extensions
- C. Object containers & configuration rules

XI. Task inputs & outputs
- A. The power of the input/output model
- B. Describing inputs and outputs
- C. Incremental builds
- D. Inferring task dependencies

XII. The Java plugin
- A. Source sets
- B. Project conventions
- C. Automated testing
- D. Building JARs

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Gradle In-Depth

Course Outline (cont'd)

XIII. The Scala plugin
- A. Project conventions
- B. Making Scala builds fast
- C. Automated testing

XIV. The Android plugin
- A. Project conventions
- B. Product Flavors
- C. Build Types
- D. Build Variants
- E. Library Projects
- F. Automated Testing
- G. Android Multi-module builds

XV. Multiproject builds
- A. Defining a multiproject build
- B. Avoiding duplicate configuration with configuration injection
- C. Inter project dependencies
- D. Downstream and upstream partial builds

XVI. The build runtime
- A. Extending all builds with init scripts
- B. Init script locations
- C. Using the Gradle wrapper
- D. Customizing the Gradle distribution with init scripts

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically