Scala-based build definition that can use the full flexibility of Scala code

Continuous compilation and testing with triggered execution

Generates documentation with scaladoc  
• Supports mixed Scala/Java projects  
• Supports testing with ScalaCheck, specs, and ScalaTest. JUnit is supported by a plugin.

Starts the Scala REPL with project classes and dependencies on the classpath

External project support (list a git repository as a dependency!)  
• Parallel task execution, including parallel test execution

Lazy val root = (project in file(“.”))

.Settings(

Name := “decisive”,

)

<https://www.packtpub.com/mapt/book/application_development/9781783282678/3/ch03lvl1sec21/resolvers>

resolvers += name at location.

For example:

resolvers += "releases" at "http://oss.sonatype.org/content/repositories/releases"

* resolvers ++= Seq (name1 at location1, name2 at location2).

For example:

resolvers ++= Seq("snapshots" at "http://oss.sonatype.org/content/repositories/snapshots",

"releases" at "http://oss.sonatype.org/content/repositories/releases")

* resolvers := Seq (name1 at location1, name2 at location2).

For example:

resolvers := Seq("sgodbillon" at "https://bitbucket.org/sgodbillon/repository/raw/master/snapshots/",

"Typesafe backup repo" at "http://repo.typesafe.com/typesafe/repo/",

"Maven repo1" at "http://repo1.maven.org/")

)

You can also add their own local Maven repository as a resource using the following syntax:

resolvers += "Local Maven Repository" at...

sbt clear console > Ctrl+L