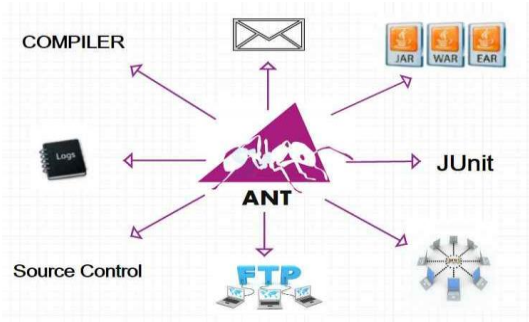
Integrate Ant project with Gradle

Ant

- Ant is a Java-based build automation tool
- It is developed to overcome the drawbacks of the Make build tool of UNIX
- It is based on procedural programming approach.



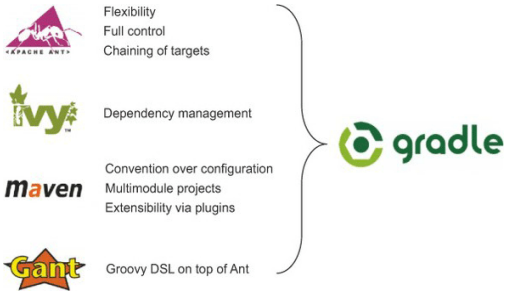
Build.xml configuration

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project default="run" name="AntWithJavaProj01 -compile and run JAVA class">
3  <target name="run" depends="compile">
4  <java classname="com.za.code.Demo" >
5  <classpath path="staging"/>
6  </java>
7  </target>
8  <target name="compile">
9  <javac  includeantruntime="false" srcdir="./src" destdir="staging"/>
10 </target>
11 </project>
```

Disadvantages of Ant

- It is an older build tool when compared to the recent tool like Gradle, Maven.
- It doesn't have a life cycle.
- It doesn't have a central repository like maven and still uses the library to store its dependencies.
- We need to provide the entire project structure in the build.

Gradle

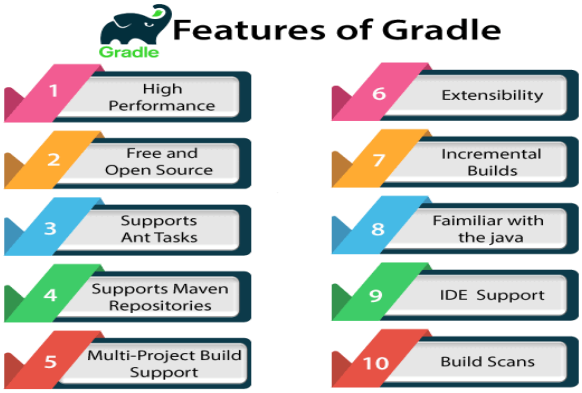
- Gradle is a Groovy-based build automation tool
- In gradle uses Groovy ,kotlin DSL(domain specific language)
- It is designed for the multi-project built
- It provides a structured build for the projects.
- It uses a DAG to define the order of executing the task.
- 

Integrate Ant project with Gradle

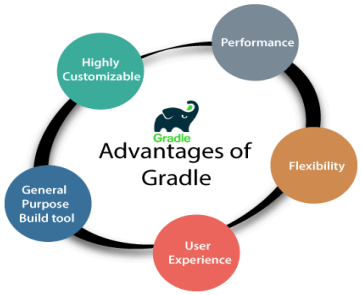
Build.gradle

```
plugins {
    id 'org.springframework.boot' version '2.6.2'
    id 'io.spring.dependency-management' version '1.0.11.RELEASE'
    id 'groovy'
}

group = 'com.example'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '1.8'

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    implementation 'org.codehaus.groovy:groovy'
    runtimeOnly 'mysql:mysql-connector-java'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
}

test {
    useJUnitPlatform()
}

```

Features of Gradle



Advantages of Gradle:



1.ant. importBuild() method

2. AntBuilder.setLifecycleLogLevel(java.lang.String) method.

steps of Using Ant from Gradle

- Using Ant tasks and types in your build
- Importing an Ant build
- Ant properties and references
- Ant logging
- API

1.Using Ant tasks and types in your build

Integrate Ant project with Gradle

You execute an Ant task by calling a method on the AntBuilder instance

Using an Ant task

GroovyKotlin

build.gradle

```
tasks.register('hello') {
    doLast {
        String greeting = 'hello from Ant'
        ant.echo(message: greeting)
    }
}
```

Output of gradle hello

```
> gradle hello

> Task :hello
[ant:echo] hello from Ant

BUILD SUCCESSFUL in 0s
1 actionable task: 1 executed
```

2.Importing an Ant build

You can use the ant.importBuild() method to import an Ant build into your Gradle project

build.gradle

```
ant.importBuild 'build.xml'
```

build.xml

```
<project>
    <target name="hello">
        <echo>Hello, from Ant</echo>
    </target>
</project>
```

3. Ant properties and references

There are several ways to set an Ant property, so that the property can be used by Ant tasks. You can set the property directly on the AntBuilder instance.

build.gradle

```
ant.buildDir = buildDir
ant.properties.buildDir = buildDir
ant.properties ['buildDir'] = buildDir
ant.properties (name: 'buildDir', location: buildDir)
```

. Getting an Ant property

build.xml

```
<property name="antProp" value="a property defined in an Ant build"/>
```

Setting an Ant reference

build.gradle

```
ant.path(id: 'classpath', location: 'libs')
ant.references.classpath = ant.path(location: 'libs')
ant.references['classpath'] = ant.path(location: 'libs')
```

build.xml

```
<path refid="classpath"/>
```

Getting an Ant reference

build.xml

```
<path id="antPath" location="libs"/>
```

build.gradle

```
println ant.references.antPath
println ant.references['antPath']
```

4.Ant logging

Ant Message Priority	Gradle Log Level
VERBOSE	DEBUG
DEBUG	DEBUG
INFO	INFO
WARN	WARN
ERROR	ERROR

5. API

The Ant integration is provided by AntBuilder.