Task :3 Gradle

Gradle Basics and Commands Documentation

Gradle is a build automation tool used for building, testing, and deploying applications. It uses a domain-specific language (DSL) based on Groovy or Kotlin for configuration.

Basic Gradle Commands

1. **./gradlew**
   - The Gradle Wrapper (gradlew) is a script that ensures the project uses the specified version of Gradle without requiring users to install Gradle manually.
   - Example:

   ```
   ./gradlew build
   ```

2. **tasks**
   - Lists available tasks in the project.
   - **--all**: Displays all tasks, including those from subprojects.
   - Example:

   ```
   ./gradlew tasks --all
   ```

3. **build**
   - Builds the project by compiling the code, running tests, and assembling outputs like .jar or .war files.
   - Example:

   ```
   ./gradlew build
   ```

4. **clean**
   - Deletes the build directory to ensure a fresh build.
   - Example:

   ```
   ./gradlew clean
   ```

5. **test**
   - Runs the unit tests for the project.
   - Example:

   ```
   ./gradlew test
   ```

6. **dependencies**
   - Displays the dependency tree of the project.
   - Example:

   ```
   ./gradlew dependencies
   ```

7. **help**
   - Provides information about the current Gradle build and help for commands.
   - Example:

```
./gradlew help
```

8. **properties**
   - Lists all properties of the current project.
   - Example:

     ```
     ./gradlew properties
     ```

Advanced and Useful Gradle Commands

1. **assemble**
   - Assembles the outputs of the project without running tests.
   - Example:

     ```
     ./gradlew assemble
     ```

2. **jar**
   - Creates a .jar file containing the compiled classes and resources.
   - Example:

     ```
     ./gradlew jar
     ```

3. **buildEnvironment**
   - Displays the build script dependencies.
   - Example:

     ```
     ./gradlew buildEnvironment
     ```

4. **components**
   - Displays the components of the project, such as artifacts or variants. *(Deprecated)*

```
./gradlew components
```

5. **dependencyInsight**
   - Shows detailed information about a specific dependency.
   - Example:

     ```
     ./gradlew dependencyInsight --dependency <dependency_name>
     ```

6. **check**

- Runs all checks, including tests and static code analysis.
- Example:

```
./gradlew check
```

7. **updateDaemonJvm**

- Updates or regenerates the Gradle Daemon JVM configuration.
- Example:

```
./gradlew updateDaemonJvm
```

8. **init**

- Initializes a new Gradle build in an empty directory.
- Example:

```
./gradlew init
```

9. **wrapper**

- Generates or updates the Gradle wrapper files for the project.
- Example:

```
./gradlew wrapper
```

10. **outgoingVariants**

- Displays the outgoing variants of a project.
- Example:

```
./gradlew outgoingVariants
```

Debugging and Configuration

1. **--stacktrace**

- Prints a detailed stack trace if an error occurs.
- Example:

```
./gradlew build --stacktrace
```

2. **--info**

- Displays additional information during the build process.
- Example:

```
./gradlew build --info
```

3. **--debug**

- Provides detailed debugging information.

- Example:

  ```
  ./gradlew build --debug
  ```

4. **--warning-mode all**
   - Shows all deprecation warnings to prepare for future Gradle versions.
   - Example:

     ```
     ./gradlew build --warning-mode all
     ```

Gradle Wrapper Commands

- **gradlew vs gradle**:
  - Use ./gradlew (Gradle Wrapper) for consistent builds, as it downloads and uses the specific Gradle version required by the project.
  - gradle (installed Gradle) uses the system-wide version, which may cause version mismatches.

Project-Specific Tasks

In your project, you discovered tasks like:

- **app:run**: Runs the project as a JVM application.
- **app:jar**: Assembles a JAR file for the app.
- **app:build**: Compiles, tests, and assembles the app.
- **app:test**: Executes the test suite for the app.

These tasks are defined in your project's build script (build.gradle) or plugins.

Common Troubleshooting Tips

1. **Command Not Recognized**
   - Ensure you are in the correct directory where the gradlew script exists.
   - Check for typos (./gradelw → ./gradlew).
2. **Outdated Gradle Version**
   - Update the wrapper with:

     ```
     ./gradlew wrapper --gradle-version <latest_version>
     ```
3. **Missing Build Scripts**
   - Ensure build.gradle or build.gradle.kts is present in the project directory.