

# Apache OFBiz®

The Apache OFBiz Project

Version Trunk

# Table of Contents

1. System requirements	2
2. Quick start	3
2.1. Download the Gradle wrapper:	3
2.2. Prepare OFBiz:	3
2.2.1. Clean system and load the complete OFBiz data	3
2.3. Start OFBiz:	3
2.4. Visit OFBiz through your browser:	3
3. Security	4
4. Build system syntax	5
4.1. Operating System Syntax	5
4.2. Types of tasks in Gradle	5
4.2.1. Example standard tasks	5
4.2.2. Example OFBiz server tasks	5
4.2.3. Example mixed tasks (standard and OFBiz server)	6
5. Quick reference	7
5.1. Help tasks	7
5.1.1. List OFBiz server commands	7
5.1.2. List build tasks	7
5.1.3. List build projects	7
5.1.4. Gradle build system help	7
5.2. Server command tasks	7
5.2.1. Start OFBiz	7
5.2.2. Shutdown OFBiz	7
5.2.3. Get OFBiz status	7
5.2.4. Force OFBiz shutdown	8
5.2.5. Start OFBiz in remote debug mode	8
5.2.6. Start OFBiz on a different port	8
5.2.7. Start OFBiz in the background	8
5.2.8. Passing JVM runtime options to OFBiz	8
5.3. Data loading tasks	8
5.3.1. Load specific OFBiz data	9
5.3.2. Load all OFBiz data	10
5.3.3. Load seed data	10
5.3.4. load ext data	10
5.3.5. load ext test data	10
5.3.6. load data from an entity file	10
5.3.7. create a new tenant	11
5.3.8. load data for a specific tenant	11

5.4. Testing tasks	11
5.4.1. Execute all unit tests	11
5.4.2. Execute all integration tests	12
5.4.3. Execute integration tests with a different log level	12
5.4.4. Execute an integration test case	12
5.4.5. Execute an integration test case in debug mode with verbose log	12
5.4.6. Execute an integration test suite	12
5.4.7. Execute an integration test suite in debug mode	12
5.5. Miscellaneous tasks	13
5.5.1. Run all tests on a clean system	13
5.5.2. Clean all generated artifacts	13
5.5.3. Refresh the generated artifacts	13
5.5.4. Create an admin user account	13
5.5.5. Compile Java without using Xlint output	13
5.5.6. Run OWASP tool to identify dependency vulnerabilities (CVEs)	13
5.5.7. Setup eclipse project for OFBiz	13
5.5.8. Package and distribute OFBiz	13
6. OFBiz plugin system	15
6.1. Pull (download and install) a plugin automatically	15
6.2. Pull an official plugin from source control	15
6.3. Pull all official plugins from source control	15
6.4. Install a plugin	16
6.5. Uninstall a plugin	16
6.6. Remove a plugin	16
6.7. Create a new plugin	16
6.8. Push a plugin to a repository	17
7. Miscellaneous Documentation	18
7.1. Further reading	18
7.2. Repository and directory structure	18
7.3. Running gradle tasks without an internet connection	18
7.4. Setup an external database like MySQL, PostgreSQL, etc	18
7.5. Setup gradle tab-completion on Unix-like systems:	19
8. Crypto notice	20

If you are reading this file in AsciiDoc format you may want to see it in [HTML](#) or [PDF](#) format

---

Welcome to **Apache OFBiz**! A powerful top level Apache software project. OFBiz is an Enterprise Resource Planning (ERP) System written in Java and houses a large set of libraries, entities, services and features to run all aspects of your business.

For more details about OFBiz please visit the OFBiz Documentation page:

[OFBiz documentation](#)

[OFBiz License](#)

*Note:* If you want to use Eclipse, read the "Setup eclipse project for OFBiz" section to set it up.

*Note:* If you want to use an external database like MySQL or PostgreSQL, read the "Setup an external database" section to set it up.

*Note:* If you want to run OFBiz without an internet connection, read the "Running gradle tasks without an internet connection" section.

*Note:* The directory structure and repositories have changed. For more information read the "Repository and directory structure" section.

# 1. System requirements

The only requirement to run OFBiz is to have the Java Development Kit (JDK) version 8 installed on your system (not just the JRE, but the full JDK) which you can download from the below link. Make sure of setting the \$JAVA\_HOME environment variable.

[JDK download](#)

[To know more about the JDK](#)

## 2. Quick start

To quickly install and fire-up OFBiz, please follow the below instructions from the command line at the OFBiz top level directory (folder).

### 2.1. Download the Gradle wrapper:

MS Windows: `init-gradle-wrapper`

Unix-like OS: `./gradle/init-gradle-wrapper`

### 2.2. Prepare OFBiz:

#### 2.2.1. Clean system and load the complete OFBiz data

*Note:* Depending on your Internet connection speed it might take a long time for this step to complete if you are using OFBiz for the first time as it needs to download all dependencies. So please be patient!

MS Windows: `gradlew cleanAll loadAll`

Unix-like OS: `./gradlew cleanAll loadAll`

### 2.3. Start OFBiz:

MS Windows: `gradlew ofbiz`

Unix-like OS: `./gradlew ofbiz`

*Note:* Ignore the % progress indicator because this task does not end as long as OFBiz is running.

### 2.4. Visit OFBiz through your browser:

[Order Back Office](#)

[Accounting Back Office](#)

[Administrator interface](#)

You can log in with the user **admin** and password **ofbiz**.



In case of problems visit our [Further reading section](#).

# 3. Security

If you find a security issue, please report it to: [security @ ofbiz.apache.org](mailto:security@ofbiz.apache.org). Once proper mitigations to the security issues are complete the OFBiz team will disclose this information to the public mailing list.

You can find more information about security in OFBiz at [Keeping OFBiz secure](#)



In production never use the credentials contained in demo data. Not only the admin credentials, but all of them.

# 4. Build system syntax

All build tasks are executed using the **Gradle** build system which is embedded in OFBiz. To execute build tasks go to OFBiz top-level directory (folder) and execute tasks from there.

## 4.1. Operating System Syntax

The syntax for tasks differ slightly between windows and Unix-like systems

- **Windows:** `gradlew <tasks-in-here>`
- **Unix-like:** `./gradlew <tasks-in-here>`

For the rest of this document, we will use the windows syntax, if you are on a Unix-like system, you need to add the `./` to gradlew

## 4.2. Types of tasks in Gradle

There are two types of tasks designed for OFBiz in Gradle:

- **Standard tasks:** To execute general standard Gradle tasks
- **OFBiz server tasks:** To execute OFBiz startup commands. These tasks start with one of the following words:
- **ofbiz** : standard server commands
- **ofbizBackground** ; server commands running in a background forked process

Tips:

- OFBiz **server commands** require "**quoting**" the commands. For example: `gradlew "ofbiz --help"`
- Shortcuts to task names can be used by writing the first letter of every word in a task name. However, you cannot use the shortcut form for OFBiz server tasks. Example: `gradlew loadAdminUserLogin -PuserLoginId=myadmin = gradlew LAUL -PuserLoginId=myadmin`
- Dependent tasks can be skipped with the `-x` switch. Example: `gradlew build -x test` does not run the tests within the build.

### 4.2.1. Example standard tasks

```
gradlew build
```

```
gradlew cleanAll loadAll testIntegration
```

### 4.2.2. Example OFBiz server tasks

```
gradlew "ofbiz --help"
```

```
gradlew "ofbiz --test" --debug-jvm
```

```
gradlew "ofbizBackground --start --portoffset 10000"
```



```
gradlew "ofbiz --shutdown --portoffset 10000"
```

```
gradlew ofbiz (default is --start)
```

### **4.2.3. Example mixed tasks (standard and OFBiz server)**

```
gradlew cleanAll loadAll "ofbiz --start --portoffset 10000"
```

# 5. Quick reference

You can use the below common list of tasks as a quick reference for controlling the system. This document uses the windows task syntax, if you are on a Unix-like system, you need to add the `./` to gradlew i.e. `./gradlew`

## 5.1. Help tasks

### 5.1.1. List OFBiz server commands

List all available commands to control the OFBiz server

```
gradlew "ofbiz --help"
```

### 5.1.2. List build tasks

List all available tasks from the build system

```
gradlew tasks
```

### 5.1.3. List build projects

List all available projects in the build system

```
gradlew projects
```

### 5.1.4. Gradle build system help

Show usage and options for the Gradle build system

```
gradlew --help
```

## 5.2. Server command tasks

### 5.2.1. Start OFBiz

```
gradlew "ofbiz --start"
```

start is the default server task so this also works:

```
gradlew ofbiz
```

### 5.2.2. Shutdown OFBiz

```
gradlew "ofbiz --shutdown"
```

### 5.2.3. Get OFBiz status

```
gradlew "ofbiz --status"
```

### 5.2.4. Force OFBiz shutdown

Terminate all running OFBiz server instances by calling the appropriate operating system kill command. Use this command to force OFBiz termination if the `--shutdown` command does not work. Usually this is needed when in the middle of data loading or testing in OFBiz.

Warning: Be careful in using this command as force termination might lead to inconsistent state / data

```
gradlew terminateOfbiz
```

### 5.2.5. Start OFBiz in remote debug mode

Starts OFBiz in remote debug mode and waits for debugger or IDEs to connect on port **5005**

```
gradlew ofbiz --debug-jvm
```

### 5.2.6. Start OFBiz on a different port

Start OFBiz of the network port offsetted by the range provided in the argument to `--portoffset`

```
gradlew "ofbiz --start --portoffset 10000"
```

### 5.2.7. Start OFBiz in the background

Start OFBiz in the background by forking it to a new process and redirecting the output to **runtime/logs/console.log**

```
gradlew "ofbizBackground --start"
```

OR

```
gradlew ofbizBackground
```

You can also offset the port, for example:

```
gradlew "ofbizBackground --start --portoffset 10000"
```

### 5.2.8. Passing JVM runtime options to OFBiz

You can pass JVM runtime options by specifying the project property `-PjvmArgs`.

```
gradlew ofbiz -PjvmArgs="-Xms1024M -Xmx2048M" -Dsome.parameter=hello
```

If you do not specify `jvmArgs`, a default of `-Xms128M -Xmx1024M` is set.

## 5.3. Data loading tasks

OFBiz contains the following data reader types:

- **seed**: OFBiz and External Seed Data - to be maintained along with source and updated whenever a system deployment is updated

- **seed-initial:** OFBiz and External Seed Data - to be maintained along with source like other seed data, but only loaded initially and not updated when a system is updated except manually reviewing each line
- **demo:** OFBiz Only Demo Data
- **ext:** External General Data (custom)
- **ext-test:** External Test Data (custom)
- **ext-demo:** External Demo Data (custom)
- **tenant:** Data to load into the master tenants database "ofbiztenant". This data is required to identify where a tenant's database is located. For more information you can review the relevant [tenant documentation](#)

Available options for the --load-data server command are the following:

- **readers=[name]:** only load data from certain readers separated by comma. e.g. seed,seed-initial,ext
- **file=[path]:** load a single file from location or several files separated by commas. e.g. /my/file/1,/my/file/2
- **dir=[path]:** load all data files found in directory
- **component=[name]:** only load data from a specific component. e.g. base
- **delegator=[name]:** use the defined delegator. Default is "default". If the value passed is "**all-tenants**" then OFBiz will load the data for all defined tenants in the system.
- **group=[name]:** override the entity group (org.apache.ofbiz). e.g. com.example.something
- **timeout=[millis]:** timeout in milliseconds
- **create-pks:** create primary keys
- **drop-pks:** drop primary keys
- **create-constraints:** create indexes and foreign keys after loading
- **drop-constraints:** drop indexes and foreign keys before loading
- **create-fks:** create dummy (placeholder) foreign keys
- **maintain-txs:** maintain timestamps in data file
- **try-inserts:** use mostly inserts
- **repair-columns:** repair column sizes (default is true w/ drop-constraints)
- **continue-on-failure:** By default OFBiz will fail and stop if it is unable to load any of the files it is attempting to load. By passing this property OFBiz will ignore failures and continue loading all files

### 5.3.1. Load specific OFBiz data

you can choose which data readers to pass in the following syntax:

```
gradlew "ofbiz --load-data readers=<readers-here-comma-separated>"
```

Example:

```
gradlew "ofbiz --load-data readers=seed,seed-initial,ext,ext-demo"
```

Beware that copying this command in Microsoft Word will automatically transform the double dash in en dashes (Unicode 0x2013: `⎓`) Other cases not related to Word were also reported. So when this command does not work check that you are using dash!

### 5.3.2. Load all OFBiz data

Loads all data sets; meant for initial loading of generic OFBiz data. Can be applied for development, testing, demonstration, etc. purposes. Be aware that executing this task can result in your data being overwritten in your database of choice.

```
gradlew loadAll OR gradlew "ofbiz --load-data"
```



Use with caution in production environments.

### 5.3.3. Load seed data

Load ONLY the seed data (not seed-initial, demo, ext\* or anything else); meant for use after an update of the code to reload the seed data as it is generally maintained along with the code and needs to be in sync for operation

```
gradlew "ofbiz --load-data readers=seed"
```

### 5.3.4. load ext data

Load seed, seed-initial and ext data; meant for manual/generic testing, development, or going into production with a derived system based on stock OFBiz where the ext data basically replaces the demo data

```
gradlew "ofbiz --load-data readers=seed,seed-initial,ext"
```

### 5.3.5. load ext test data

Load seed, seed-initial, ext and ext-test data; meant for automated testing with a derived system based on stock OFBiz

```
gradlew "ofbiz --load-data readers=seed,seed-initial,ext,ext-test"
```

### 5.3.6. load data from an entity file

Load data from an XML file holding entity data.

```
gradlew "ofbiz --load-data file=foo/bar/FileNameHere.xml"
```

### 5.3.7. create a new tenant

Create a new tenant in your environment, create the delegator, load initial data with admin-user and password (needs multitenant=Y in general.properties). The following project parameters are passed:

- tenantId: mandatory
- tenantName: optional, default is value of tenantId
- domainName: optional, default is org.apache.ofbiz
- tenantReaders: optional, default value is seed,seed-initial,demo
- dbPlatform: optional, D(Derby), M(MySQL), O(Oracle), P(PostgreSQL) (default D)
- dbIp: optional, ip address of the database
- dbUser: optional, username of the database
- dbPassword: optional, password of the database

```
gradlew createTenant -PtenantId=mytenant
```

```
gradlew createTenant -PtenantId=mytenant -PtenantName="My Name" -PdomainName=com.example  
-PtenantReaders=seed,seed-initial,ext -PdbPlatform=M -PdbIp=127.0.0.1 -PdbUser=mydbuser  
-PdbPassword=mydbpass
```

If run successfully, the system creates a new tenant having:

- delegator: default# $\{tenantId\}$  (e.g. default#mytenant)
- admin user:  $\{tenantId\}$ -admin (e.g. mytenant-admin)
- admin user password: ofbiz

### 5.3.8. load data for a specific tenant

Load data for one specific tenant in a multitenant environment. Note that you must set multitenant=Y in general.properties and the following project parameters are passed:

- tenantId (mandatory)
- tenantReaders (optional)
- tenantComponent (optional)

```
gradlew loadTenant -PtenantId=mytenant
```

```
gradlew loadTenant -PtenantId=mytenant -PtenantReaders=seed,seed-initial,demo  
-PtenantComponent=base
```

## 5.4. Testing tasks

### 5.4.1. Execute all unit tests

```
gradlew test
```

## 5.4.2. Execute all integration tests

```
gradlew testIntegration
```

OR

```
gradlew 'ofbiz --test'
```

## 5.4.3. Execute integration tests with a different log level

It is possible to start integration tests with a log level different from the default one. The log levels allowed are listed below from most verbose to least verbose:

- always
- verbose
- timing
- info
- important
- warning
- error
- fatal

```
gradlew "ofbiz --test loglevel=fatal"
```

## 5.4.4. Execute an integration test case

run a test case, in this example the component is "entity" and the case name is "entity-tests"

```
gradlew "ofbiz --test component=entity --test suiteName=entitytests --test case=entity-query-tests"
```

## 5.4.5. Execute an integration test case in debug mode with verbose log

listens on port 5005

```
gradlew "ofbiz --test component=entity --test loglevel=verbose" --debug-jvm
```

## 5.4.6. Execute an integration test suite

```
gradlew "ofbiz --test component=entity --test suiteName=entitytests"
```

## 5.4.7. Execute an integration test suite in debug mode

listens on port 5005

```
gradlew "ofbiz --test component=entity --test suiteName=entitytests" --debug-jvm
```

## 5.5. Miscellaneous tasks

### 5.5.1. Run all tests on a clean system

```
gradlew cleanAll loadAll testIntegration
```

### 5.5.2. Clean all generated artifacts

```
gradlew cleanAll
```

### 5.5.3. Refresh the generated artifacts

```
gradlew clean build
```

### 5.5.4. Create an admin user account

Create an admin user with login name MyUserName and default password with value "ofbiz". Upon first login OFBiz will request changing the default password

```
gradlew loadAdminUserLogin -PuserLoginId=MyUserName
```

### 5.5.5. Compile Java without using Xlint output

By default Xlint prints output of all warnings detected by the compiler, if you want to silence them

```
gradlew -PXlint:none build
```

### 5.5.6. Run OWASP tool to identify dependency vulnerabilities (CVEs)

The below command activates a gradle plugin (OWASP) and Identifies and reports known vulnerabilities (CVEs) in OFBiz library dependencies. The task takes time to complete, and once done, a report will be generated in \$OFBIZ\_HOME/build/reports/dependency-check-report.html

```
gradlew -PenableOwasp dependencyCheckAnalyze
```

### 5.5.7. Setup eclipse project for OFBiz

Setting up OFBiz on eclipse is done by simply running the below command and then importing the project to eclipse. This command will generate the necessary **.classpath** and **.project** files for eclipse and it will also make the source code for external libraries available in eclipse (i.e. you can view source through Ctrl + Click)

The first time you run this command it will take a long time to execute because it will download source packages available for project dependencies.

```
gradlew eclipse
```

### 5.5.8. Package and distribute OFBiz

In order to deploy OFBiz on a target system and in particular in a production environment without requiring the target system to download Gradle and OFBiz dependencies from the internet, it is



possible to generate an archive bundling OFBiz with all the Jars it depends on as a `tar` archive

```
gradlew distTar
```

or as a `zip` archive.

```
gradlew distZip
```

Those archives are available in the `build/distributions` directory. To run OFBiz from those archive you must first unarchive them with `tar xf` or `unzip` and then from that directory you can run either `bin/ofbiz` shell script or `bin/ofbiz.bat` batch script with the appropriate ofbiz options.

## 6. OFBiz plugin system

OFBiz provides an extension mechanism through plugins. Plugins are standard OFBiz components that reside in the plugins directory. Plugins can be added manually or fetched from a maven repository. The standard tasks for managing plugins are listed below.

*Note:* OFBiz plugin versions follow [Semantic Versioning 2.0.0](#)

### 6.1. Pull (download and install) a plugin automatically

Download a plugin with all its dependencies (plugins) and install them one-by-one starting with the dependencies and ending with the plugin itself.

```
gradlew pullPlugin -PdependencyId="org.apache.ofbiz.plugin:myplugin:0.1.0"
```

If the plugin resides in a custom maven repository (not jcenter or localhost) then you can use specify the repository using below command:

```
gradlew pullPlugin -PrepoUrl="http://www.example.com/custom-maven" -PdependencyId="org.apache.ofbiz.plugin:myplugin:0.1.0"
```

If you need username and password to access the custom repository:

```
gradlew pullPlugin -PrepoUrl="http://www.example.com/custom-maven" -PrepoUser=myuser -PrepoPassword=mypassword -PdependencyId="org.apache.ofbiz.plugin:myplugin:0.1.0"
```

### 6.2. Pull an official plugin from source control

Download an official plugin from source control (currently subversion) and place it in the plugins directory. In addition, this task also executes the "install" task if it is defined for the plugin being downloaded.

This task is mostly useful when working on the trunk branch as it pulls in the latest version of a plugin.

```
gradlew pullPluginSource -PpluginId=ecommerce
```

*Note:* This plugin will have its own .svn directory placed inside the plugin directory.

### 6.3. Pull all official plugins from source control

Download all officially supported plugins from source control (currently subversion) and place them inclusive of their ".svn" directory in /plugins. WARNING! This task deletes the /plugins directory and replaces it with the official plugins.

```
gradlew pullAllPluginsSource
```

This task makes it easy to download and develop officially supported plugins. It is mostly used by

developers or individuals working on the trunk branch. We do not recommend using this task on releases of OFBiz, instead consider using the "pullPlugin" task to get the correct version of a plugin compatible with your release.

*Note:* All the plugins will share a .svn directory placed in the plugins directory.

## 6.4. Install a plugin

If you have a plugin called mycustomplugin and want to install it in OFBiz follow the below instructions:

- Extract the plugin if it is compressed
- Place the extracted directory into /plugins
- Run the below command

```
gradlew installPlugin -PpluginId=myplugin
```

The above commands executes the task "install" in the plugin's build.gradle file if it exists

## 6.5. Uninstall a plugin

If you have an existing plugin called mycustomplugin and you wish to uninstall run the below command

```
gradlew uninstallPlugin -PpluginId=myplugin
```

The above command executes the task "uninstall" in the plugin's build.gradle file if it exists

## 6.6. Remove a plugin

Calls **uninstallPlugin** on an existing plugin and then delete it from the file-system

```
gradlew removePlugin -PpluginId=myplugin
```

## 6.7. Create a new plugin

Create a new plugin. The following project parameters are passed:

- pluginId: mandatory
- pluginResourceName: optional, default is the Capitalized value of pluginId
- webappName: optional, default is the value of pluginId
- basePermission: optional, default is the UPPERCASE value of pluginId

```
gradlew createPlugin -PpluginId=myplugin
```

```
gradlew createPlugin -PpluginId=myplugin -PpluginResourceName=MyPlugin -PwebappName=mypluginweb  
-PbasePermission=MYSECURITY
```

The above command creates a new plugin in /plugins/myplugin

## 6.8. Push a plugin to a repository

This task publishes an OFBiz plugin into a maven package and then uploads it to a maven repository. Currently, pushing is limited to localhost maven repository (work in progress). To push a plugin the following parameters are passed:

- pluginId: mandatory
- groupId: optional, defaults to org.apache.ofbiz.plugin
- pluginVersion: optional, defaults to 0.1.0-SNAPSHOT
- pluginDescription: optional, defaults to "Publication of OFBiz plugin \${pluginId}"

```
gradlew pushPlugin -PpluginId=myplugin
```

```
gradlew pushPlugin -PpluginId=mycompany -PpluginGroup=com.mycompany.ofbiz.plugin  
-PpluginVersion=1.2.3 -PpluginDescription="Introduce special functionality X"
```

# 7. Miscellaneous Documentation

## 7.1. Further reading

- [Known issues](#)

## 7.2. Repository and directory structure

OFBiz is split into two repositories:

- **ofbiz-framework**: Contains the core framework and main applications in the system like accounting, party, order, etc
- **ofbiz-plugins**: Renamed from "special-purpose" and contains optional components that are officially supported by the community

Furthermore, the hot-deploy directory is removed as the plugins directory works as a replacement for both "special-purpose" and "hot-deploy".

If you need to load the components in the plugins directory in a specific order place a component-load.xml file in the plugins directory listing the order.

To check out a plugin from source control use the **pullPluginSource** Gradle task. To check out all plugins from source control use the **pullAllPluginsSource**. **Beware** this deletes a previously existing plugins directory.

## 7.3. Running gradle tasks without an internet connection

OFBiz must run with an internet connection the **first time** it is prepared on the system because it needs to download all the required dependencies.

After preparing OFBiz the first time correctly, it is possible to run OFBiz without an internet connection by using the **--offline** command line switch which tells Gradle to fetch its dependencies from the cache.

If any dependencies are missing from the cache and you pass **--offline** switch then the build execution will fail.

## 7.4. Setup an external database like MySQL, PostgreSQL, etc

To setup an external database instead of the default embedded Apache Derby, you will need to follow the following instructions:

1. Find the JDBC driver suitable for your database using one of the following options:

- Search for the JDBC driver in [jcenter](#) and place it in build.gradle dependencies e.g. `runtime 'mysql:mysql-connector-java:5.1.36'`

OR

- Download the JDBC driver jar and place it in `$OFBIZ_HOME/lib` or the lib sub-directory of any component
2. Modify the entityengine.xml file located in `$OFBIZ_HOME/framework/entity/config` to switch the default database to the one you selected. For more details you can read the relevant section in the [technical setup guide](#)

## 7.5. Setup gradle tab-completion on Unix-like systems:

To get tab completion (auto complete gradle commands by pressing tab) you can download the script from the below link and place it in the appropriate location for your system.

### [Gradle tab completion](#)

For example, on debian based systems, you can use the following command:

```
sudo curl -L -s https://edub.me/gradle-completion-bash -o /etc/bash_completion.d/gradle-tab-completion.bash
```

## 8. Crypto notice

This distribution includes cryptographic software. The country in which you currently reside may have restrictions on the import, possession, use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check your country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted. See <http://www.wassenaar.org/> for more information.

The U.S. Government Department of Commerce, Bureau of Industry and Security (BIS), has classified this software as Export Commodity Control Number (ECCN) 5D002.C.1, which includes information security software using or performing cryptographic functions with asymmetric algorithms. The form and manner of this Apache Software Foundation distribution makes it eligible for export under the License Exception ENC Technology Software Unrestricted (TSU) exception (see the BIS Export Administration Regulations, Section 740.13) for both object code and source code.

The following provides more details on the included cryptographic software:

- Various classes in OFBiz, including DesCrypt, HashCrypt, and BlowFishCrypt use libraries from the Sun Java JDK API including `java.security.*` and `javax.crypto.*` (the JCE, Java Cryptography Extensions API)
- Other classes such as `HttpClient` and various related ones use the JSSE (Java Secure Sockets Extension) API