


From Ant to Gradle - R16.11 version

 **Deprecated**
This is quite deprecated now (2024/02/05)

 **If you are new to OFBiz or come from an OFBiz version which uses Ant for building and now want to use a newer version with Gradle, here is the documentation you are looking for.**

There is also a [newer page for the trunk](#)

- [Step-by-step guide](#)
- [Gradle documentation](#)

- Apache OFBiz®
- [System requirements](#)
 - [Security](#)
 - [Quick start](#)
 - [Build system syntax](#)
 - [Quick reference](#)
 - [OFBiz plugin system](#)
 - [Useful Tips](#)
 - [Crypto notice](#)
 - [Related articles](#)

Step-by-step guide

1. Though it's part of the CleanAll Gradle task, you might be interested by the temporary cleanAnt task which cleans old artifacts generated by Ant. This task has been removed from the trunk but is still present in the R16.11 branch hence all R16.11.* releases
2. If you used to run one of the download-* Ant targets (download-PG-JDBC, download-activemq, download-mysql-JDBC) those have been removed.

See

 Unable to render Jira issues macro, execution error.

if interested in more details

3. If you use your own custom libs (jar) shared by several of your own components you might put these libs in the lib folder just under the OFBiz root folder.

You will find a README there with more details. You might also read

 Unable to render Jira issues macro, execution error.

fo

r history reason.

4. It's clearly documented in the README.MD and below, but you might be interested to know that the debug port is now the standard 5005 rather than previously the OFBiz specific 8091
5. If you are interested by all the changes done in the context of the Ant to Gradle migration please check

 Unable to render Jira issues macro, execution error.

Gradle documentation

The documentation bloc below is the HTML import of the conversion of [README.MD](#) into HTML by [Pandoc](#) using this command line locally in root folder

```
pandoc README.md -s -o README.md.html
```

Here is [README.md.html](#) in the repository that you are reading below

Apache OFBiz®

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](#)

System requirements

The first 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.

[JDK download](#)

[To know more about the JDK](#)

The second requirement is to setup the Gradle wrapper files that are used to build the system. In order to fulfill this requirement, you can run the following command:

MS Windows: `init-gradle-wrapper`

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

The script will download all the Gradle wrapper files required to build the system. Alternatively, after you have Gradle installed:

[Installing Gradle](#)

you can setup the Gradle wrapper for OFBiz by executing the following from the command line at the OFBiz top level directory (folder):

```
gradle wrapper --gradle-version 2.13
```

Note: if you are using Eclipse, make sure of running the appropriate Eclipse command `gradlew eclipse` before creating the project in Eclipse. This command will prepare OFBiz for Eclipse with the correct classpath and settings by creating the `.classpath` and `.project` files.

Security

You can trust the OFBiz Project Management Committee members and committers do their best to keep OFBiz secure from external exploits, and fix vulnerabilities as soon as they are known. Despite these efforts, if ever you find and want to report a security issue, please report at: [security @ ofbiz. apache.org](mailto:security@ofbiz.apache.org), before disclosing them in a public forum.

Note: Be sure to read this Wiki page if ever you plan to use RMI, JNDI, JMX or Spring and maybe other Java classes OFBiz does not use Out Of The Box (OOTB): [The infamous Java serialization vulnerability](#)

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

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)

Prepare OFBiz:

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 loadDefault`

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

Start OFBiz:

MS Windows: `gradlew ofbiz`

Unix-like OS: `./gradlew ofbiz`

Visit OFBiz through your browser:

[Order Back Office](#)

[Accounting Back Office](#)

[Administrator interface](#)

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

Note: the default configuration uses an embedded Java database (Apache Derby) and embedded application server components such as Apache Tomcat®, Apache Geronimo (transaction manager), etc.

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.

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

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
 - **ofbizDebug** : server commands running in remote debug mode
 - **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`

Example standard tasks

```
gradlew build
```

```
gradlew cleanAll loadDefault testIntegration
```

Example OFBiz server tasks

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

```
gradlew "ofbizDebug --test"
```

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

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

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

Example mixed tasks (standard and OFBiz server)

```
gradlew cleanAll loadDefault "ofbiz --start"
```

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`

Help tasks

List OFBiz server commands

List all available commands to control the OFBiz server

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

List build tasks

List all available tasks from the build system

```
gradlew tasks
```

List build projects

List all available projects in the build system

```
gradlew projects
```

Gradle build system help

Show usage and options for the Gradle build system

```
gradlew --help
```

Server command tasks

Start OFBiz

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

start is the default server task so this also works:

```
gradlew ofbiz
```

Shutdown OFBiz

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

Get OFBiz status

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

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
```

Start OFBiz in remote debug mode

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

```
gradlew "ofbizDebug --start"
```

OR

```
gradlew ofbizDebug
```

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"
```

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"
```

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.

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)

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"
```

Load default OFBiz data

Loads default data set; 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. Use with caution in production environments. The default data set is defined by datasource using the `read-data` attribute, followed by the name of the data set, into the `datasource` element of the `'entityengine.xml'` file.

```
gradlew loadDefault
```

OR

```
gradlew "ofbiz --load-data"
```

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"
```

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"
```

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"
```

load data from an entity file

Load data from an XML file holding entity data.

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

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

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
```

Testing tasks

Execute all unit tests

```
gradlew test
```

Execute all integration tests

```
gradlew testIntegration
```

OR

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

Execute an integration test case

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

```
gradlew "ofbiz --test component=entity --test case=entity-tests"
```

Execute an integration test case in debug mode

listens on port **5005**

```
gradlew "ofbizDebug --test component=entity --test case=entity-tests"
```

Execute an integration test suite

```
gradlew "ofbiz --test component=widget --test suiteName=org.apache.ofbiz.widget.test.WidgetMacroLibraryTests"
```

Execute an integration test suite in debug mode

listens on port **5005**

```
gradlew "ofbizDebug --test component=widget --test suite=org.apache.ofbiz.widget.test.WidgetMacroLibraryTests"
```

Miscellaneous tasks

Launch a graphical user interface of Gradle

This is a very convenient feature of Gradle which allows the user to interact with Gradle through a swing GUI. You can save frequently used commands in a list of favorites for frequent reuse.

```
gradlew --gui
```

Run all tests on a clean system

```
gradlew cleanAll loadDefault testIntegration
```

Clean all generated artifacts

```
gradlew cleanAll
```

Refresh the generated artifacts

```
gradlew clean build
```

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
```

Compile Java using Xlint output

Xlint prints output of all warnings detected by the compiler

```
gradlew -PXlint build
```

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. This command takes a long time to execute because it needs to download all plugin dependencies and the CVE identification process is also time consuming

```
gradlew -PenableOwasp dependencyCheck
```

Setup eclipse project for OFBiz

Thanks to some gradle magic, setting up OFBiz on eclipse is very easy. All you need is to execute one command and then you can import the project to eclipse. This command will generate the necessary **.classpath** and **.project** files for eclipse.

```
gradlew eclipse
```

OFBiz plugin system

OFBiz provides an extension mechanism through plugins. Plugins are standard OFBiz components that reside in the specialpurpose 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](#)

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"
```

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 /specialpurpose
- Run the below command

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

The above commands achieve the following:

- add the plugin to /specialpurpose/component-load.xml
- executes the task "install" in the plugin's build.gradle file if it exists

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 commands achieve the following:

- executes the task "uninstall" in the plugin's build.gradle file if it exists
- removes the plugin from /specialpurpose/component-load.xml

Remove a plugin

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

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

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 commands achieve the following:

- create a new plugin in /specialpurpose/myplugin
- add the plugin to /specialpurpose/component-load.xml

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"
```

Useful Tips

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://gist.github.com/nolanlawson/8694399/raw/gradle-tab-completion.bash -o /etc/bash_completion.d/gradle-tab-completion.bash
```

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

Related articles

Gradle Plugins that could be used to improve OFBiz code

Style

https://docs.gradle.org/current/userguide/checkstyle_plugin.html

Bugs

https://docs.gradle.org/current/userguide/findbugs_plugin.html

Better code

https://docs.gradle.org/current/userguide/pmd_plugin.html

https://docs.gradle.org/current/userguide/jdepend_plugin.html

https://docs.gradle.org/current/userguide/codenarc_plugin.html

Tests coverage

<https://plugins.gradle.org/plugin/org.sonarqube>

https://docs.gradle.org/current/userguide/jacoco_plugin.html