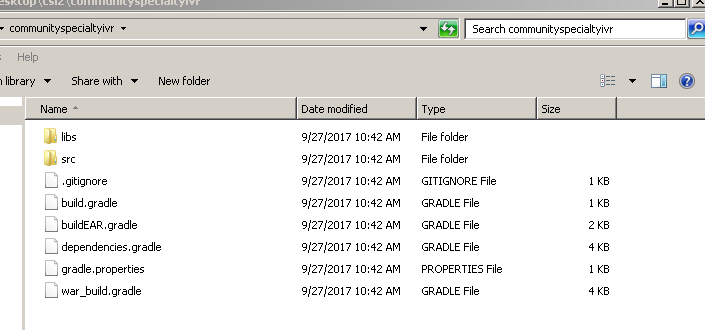
# Gradle for IVR projects

When you first clone a gradle project it will/should be missing the .classpath and .project files proper for eclipse projects, because of this the project cannot be recognized or imported into RAD or Eclipse.

These files are not included and must not be committed as they have references to local paths.



Two main gradle files have been created with different purposes:

1. **build.gradle** is the default build file for gradle and will be used to **generate the eclipse project** . (this file is used by default if other is nor provided)
2. **buildEAR.gradle** includes the ear task and is used to **generate the EAR** file for deployment, (you have to provide in command line when you want to use this file)

Two secondary files where created:

1. **dependencies.gradle**: is the file where all the dependencies have to be declared ( **is very important to put dependencies in this file as it is used by the two main gradle files**)
2. **war\_build.gradle**: this file contains all the logic to update the files according to the environment provided, the logic to leave local files as they were before updating and the war task to package the war file.

## Generate Eclipse project from gradle

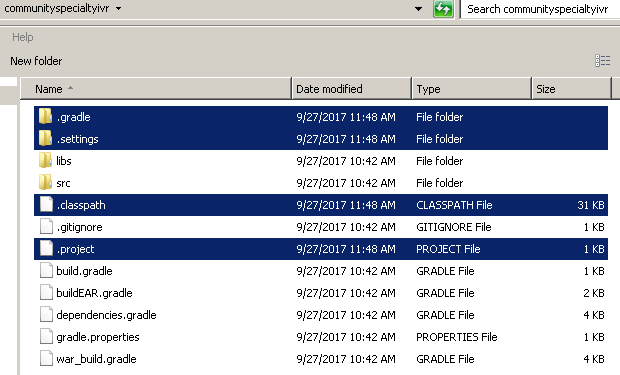
1. Having correctly set the your gradle bin path into Windows PATH using Windows console, ex :

set PATH=%PATH%;C:\gradle-4.0.2-bin\gradle-4.0.2\bin

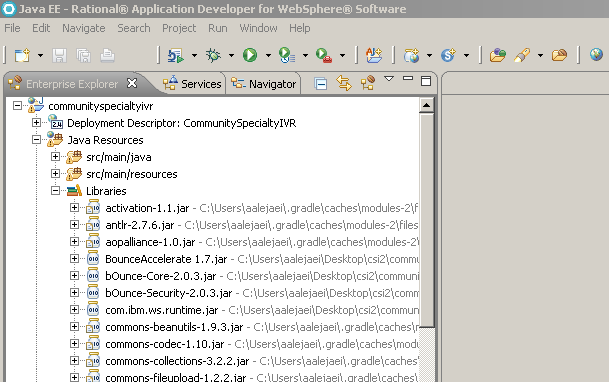
1. Move to the folder of your project where the **build.gradle** file is and run:

gradle -Dhttps.proxyHost=172.28.97.24 -DproxyPort=8080 eclipse

1. This will generate the project files needed to import the project into Eclipse and implement the proxy required to download dependencies.



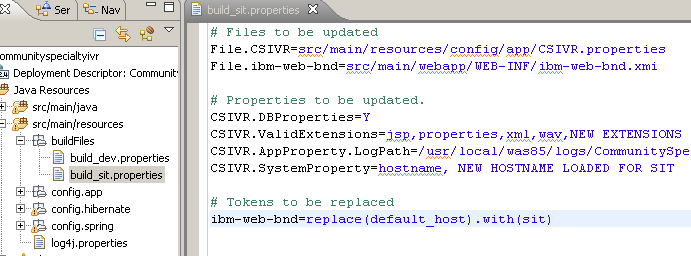
1. Now you can import the project into eclipse:



## Configure build files for different environments

The build files have to be placed under src/main/resources/buildFiles/ using the naming pattern **build\_xxxxx.properties,** where xxxxx is the environment that hast to be provided in the command line.

Below example for **build\_sit.properties**



The build properties files are divided into three parts

1. **Files to be updated**: contains the list of files and their paths that the script has to change when creating the EAR file (they can be **properties files or xml/text files**).

File.**CSIVR**=src/main/resources/config/app/CSIVR.properties

File.**ibm-web-bnd**=src/main/webapp/WEB-INF/ibm-web-bnd.xmi

**(**name used for mapping with properties marked in black**)**

1. **Properties to be updated**: this part only contains properties to be updated **into** **properties files**:

**CSIVR**.DBProperties=Y

**CSIVR**.ValidExtensions=jsp,properties,xml,wav,NEW EXTENSIONS

**(**name marked in black has to match a defined file in the Files to be updated section**)**

1. **Tokens to be replaced** in xml or any text files, it will replace all the occurrences of what is inside **replace ( )** with the characters that are inside **with( )**:

**ibm-web-bnd**=replace(default\_host).with(sit)

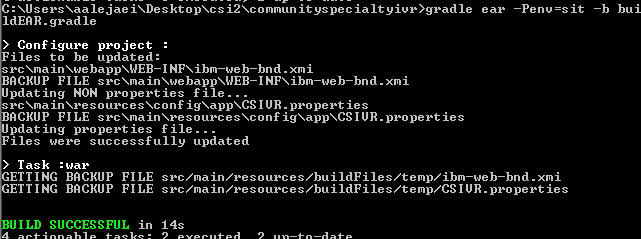
**(**name marked in black has to match a defined file in the Files to be updated section**)**

**(**this entry will make the script find all the occurrences of **“default\_host” and replace them with “sit”** in the file **ibm-web-bnd**  **)**

## Generate EAR file

To generate the EAR file its required to provide the **environment** and the **buildEAR.gradle** file in the command line:

gradle ear -Penv=**sit** -b **buildEAR.gradle**

****

This script will back up your original files in a temporary folder, then it will replace the properties and tokens from build\_sit.properties and at the will place back your original files.