

Step 1: Assembling BuildTools

1. Download BuildTools from mineacademy.org/buildtools
2. Place BuildTools.jar into a separate folder. Copy over the launching script from your test server there and edit the first line with java parameters. It should look like this:

Type your text

```
java -Xms2G -Xmx2G -jar BuildTools.jar --rev latest --remapped  
pause
```

Take note for the `--remapped` argument, this will install additional Spigot jars into your local Maven repository with the classifiers “remapped-obf” and “remapped-mojang” so you can use it in your pom.xml file for projects.

3. Launch the script and wait until it says
“Success! Everything completed successfully. Copying final .jar files now.”



Step 2: Updating pom.xml

1. Open mineacademy.org/mappings-pom
2. Remove any Minecraft versions 1.17 or newer from your pom.xml as this will cause conflicts.
3. Update your pom.xml according to the instructions in the paste above.

```
XML 2.13 KB | None | 0 0 raw download clone embed print report
1. ....
2. Place the following code in your <properties> section.
3. ....
4.
5. <!-- Change with the most recent version you want to use for Mojangs mappings. -->
6. <remapped.version>1.19.2-R0.1-SNAPSHOT</remapped.version>
7.
8.
9. ....
10. Place the following code in your <dependencies> section.
11. ....
12.
13. <dependency>
14.     <groupId>org.spigotmc</groupId>
15.     <artifactId>spigot</artifactId>
16.     <version>${remapped.version}</version>
17.     <scope>provided</scope>
18.     <classifier>remapped-mojang</classifier>
19. </dependency>
20.
21.
22. ....
23. Place the following code in your <plugins> section in <build>.
24. ....
25.
```



Step 3: Fixing compile errors and testing

1. I recommend commenting out all NMS dependencies from your pom.xml during update process to avoid accidental imports.
2. Duplicate your NMS classes.
3. Since Mojang's names are very different from Spigot's, use mineacademy.org/mappings to help you find the new names.
4. Hot-swap / live debug only works with Spigot's mapping and not here since Ant (build.xml) doesn't support remapping. You will have to "clean install" each time using Maven and start your server manually. Give it a good test!

NMSMapper [Version overview](#) [Documentation](#)

Mojang/Spigot comparison

Only symbols which had a Spigot mapping in one of the compared versions are shown.

Classes which were not present in version 1.16.5 are filtered out, because their Spigot mapping is probably the same as their Mojang mapping.

For searching use a built-in tool in your browser (Ctrl+F or F3)

	1.16.5	1.17.1
Class:		
Mojang	com.mojang.math.Matrix3f	com.mojang.math.Matrix3f
Spigot	net.minecraft.server.v1_16_R3.Matrix3f	com.mojang.math.Matrix3f
Class:		
Mojang	com.mojang.math.Matrix4f	com.mojang.math.Matrix4f
Spigot	net.minecraft.server.v1_16_R3.Matrix4f	com.mojang.math.Matrix4f
Class:		
Mojang	com.mojang.math.OctahedralGroup	com.mojang.math.OctahedralGroup
Spigot	net.minecraft.server.v1_16_R3.PointGroupO	com.mojang.math.PointGroupO
Fields:		
Mojang	INVERT_Y	INVERT_Y
Spigot	INVERT_Y	
Mojang	INVERT_Z	INVERT_Z
Spigot	INVERT_Z	
Mojang	ROT_60_REF_NNN	ROT_60_REF_NNN
Spigot	ROT_60_REF_NNN	
Mojang	ROT_60_REF_NNP	ROT_60_REF_NNP
Spigot	ROT_60_REF_NNP	
Mojang	ROT_60_REF_NPN	ROT_60_REF_NPN
Spigot	ROT_60_REF_NPN	
Mojang	ROT_60_REF_NPP	ROT_60_REF_NPP



Multi-version support

- **A) Reflection (requires Mojang's mappings)** — To support multiple Minecraft versions with different mappings, you can sometimes just stay on the most recent mappings and use reflection to call methods that were different in older MC versions.

You will need to downgrade your MC version to an older one using older BuildTools revision (`--rev` parameter) and then update `pom.xml` with it, see what breaks, patch it using reflection, and then release your plugin.

- **B) Multiple projects (supports Spigot's mappings)** — For large code changes you will need to create the following setup:
 1. An interface project
 2. Minecraft 1.19 NMS project
 3. Minecraft 1.20 NMS project
 4. Your main plugin

Your main plugin will have 1, 2, 3 as dependencies and shade them. Your interface project will only include Foundation (or nothing). Your two NMS projects will only include your interface project and Foundation. Keep things simple, you do not even need to use “parent” mechanism in `pom.xml` to achieve this.

