

Development & Beta Testing

Information regarding how to configure beta testing environments

- [Configuring an IntelliJ Beta-testing Environment From Scratch](#)
- [Exporting your Work](#)

Configuring an IntelliJ Beta-testing Environment From Scratch

WIP - Proper formatting, better explanations, and images to be added later (but this should do for now)

Preparation:

[Download](#) and install IntelliJ (Community Edition).

[Download](#) and install Java 8.

Setting up IntelliJ:

1. Clone the Zora no Densha repository into an empty folder.
2. Using either the command line or a Git management tool (such as GitKraken), switch the branch of the downloaded repository to **mc-1.7.10**.
3. Start IntelliJ and open the downloaded repository's folder as the current project. Allow IntelliJ to 'trust' the folder/project if such a warning appears.
4. Allow the project to automatically configure itself. There should be a progress bar or status message within IntelliJ advising that the project is currently importing.
5. Once the project is fully imported and no other tasks are running in the background, add a Run Configuration to IntelliJ to be able to start the game. A **Gradle** run configuration should be added with the task **runClient**.
6. Change Gradle Java version to 1.8 (elephant on the right, then the small cogwheel)
7. Start the game using the **Debug** function in IntelliJ.

Exporting your Work

When you finished implementing your feature/fix and want to export/test your work you need to use one of these commands.

Setting up Gradle run configs:

Command	Effect
runClient/runServer	Starts the game locally without exporting to a .jar file
build	Exports the current state of the mod to a .jar mod file