

Building on the server

Building JIPipe from Source on a Linux Server using MobaXterm

This guide explains how to build and run JIPipe from source code on a Linux server using **MobaXterm** and **IntelliJ Remote Development**.

1. Requirements

JIPipe requires:

- Java
- Maven
- Git

Maven is already installed on the **Chrysos** server at the moment.

If Maven or Java is missing on another server, contact **Ruman** for installation support.

All steps below are performed through **MobaXterm**.

2. Connect to the Server using MobaXterm

1. Open **MobaXterm**
2. Click **Session**
3. Select **SSH**
4. Enter:
 - **Remote host:** your server hostname or IP
 - **Specify username:** your Linux username

5. Click **OK**

You should now have a Linux terminal inside MobaXterm.

3. Clone the JIPipe Repository

Clone the repository from the internal ASB GitLab server:

```
git clone https://asb-git.hki-jena.de/RGerst/jipipe
```

Go into the repository:

```
cd jipipe
```

You can either:

- build directly on `master`
- or switch to your own branch

Example:

```
git checkout my-branch
```

4. Recommended: Use IntelliJ Remote Development

We strongly recommend using **IntelliJ IDEA Remote Development** for building JIPipe on remote servers.

Benefits:

- easier Maven handling
- easier indexing
- integrated terminal
- easier debugging
- easier remote GUI support

Open the remote project in IntelliJ and allow Maven indexing to finish before building.

5. Install Missing Maven Dependencies (Very Important)

Some Maven dependencies are no longer available online and must be copied manually.

Download or copy the dependency files from:

https://asb-git.hki-jena.de/RGerst/jipipe-distribution-files/-/tree/main/scripts/maven-dependencies?ref_type=heads

These dependencies must be copied into the local Maven repository:

```
~/.m2/repository
```

The `.m2` folder is hidden.

To locate it:

```
cd ~  
ls -A
```

You should see:

```
.m2
```

Copy the dependency folders into:

```
~/m2/repository
```

“ ⚠ This step is extremely important.
Without these dependencies the Maven build may fail.

You can use the MobaXterm file browser (left side panel) to drag-and-drop the dependency folders into the correct location.

6. Build JIPipe

Inside the repository root:

```
cd ~/jipipe
```

Run a full Maven install:

```
mvn clean install
```

This may take a long time during the first build.

Wait until all JAR files are built successfully.

7. Run JIPipe from Source

After the build finishes, start JIPipe using:

```
java --add-opens=java.base/java.lang=ALL-UNNAMED -cp "$(cat jipipe-launcher/cp.txt):jipipe-launcher/target/classes" org.hkijena.jipipe.launcher.JIPipeLauncher
```

Example terminal location:

```
(base) dkhatri@chrysos:~/jipipe$
```

If everything was built correctly, JIPipe should start normally.

For remote GUI usage, we recommend using **MobaXterm X11 forwarding**.

Before connecting:

1. Open **MobaXterm**
2. Go to:
 - **Settings** → **X11**
3. Ensure:
 - **X11 server is enabled**

When creating the SSH session:

1. Open **Session**
2. Select **SSH**
3. Enable:
 - **X11-Forwarding**

Then launch JIPipe normally.

The JIPipe GUI should appear on your Windows desktop.

8. Troubleshooting

Missing Dependencies

If Maven reports missing artifacts:

- verify that all manually provided dependencies were copied into:

```
~/.m2/repository
```

- restart IntelliJ Maven indexing if necessary
-

Permission Issues

If scripts cannot be executed:

```
chmod +x *.sh
```

Maven Not Found

Check Maven installation:

```
mvn --version
```

If the command is not found, contact **Ruman**.

Revision #2

Created 2026-05-13 12:35:20 UTC by Dhruv Khatri

Updated 2026-05-13 12:42:16 UTC by Dhruv Khatri