

General Development Environment Setup Guide

This page is an attempt at a general style of help for users. Since there is an essentially infinite set of tool combinations to use, we will supply a guide for each tool, written by an IHMC developer who uses that tool. Some pages may be a work in progress.

Remember to follow the guidelines for [getting help](#).

For a lean quickstart guide, reference [Java Environment Setup Quickstart](#)

Contents

- [Getting Started](#)
- [Software Development](#)
- [General Troubleshooting](#)
 - [System and application crashes - GC Overhead Exceptions](#)

Getting Started

Remember to **read** and adhere to the linked Confluence documentation for each tool. We have specific ways of using these tools and installation and configuration are important.

In general:

1. Pick an operating system

[Ubuntu](#) - Free, most compatible with robots

[Windows](#) - Requires setting up [long paths](#) and installing [redistributables](#).

[MacOS](#)

2. Install all of the following: (follow installation guides in linked pages)

a. Graphics driver. For Nvidia GPUs and SCS 1, the proprietary Nvidia driver is currently required. Otherwise you will get the error "Could not find GLX 1.3 config from peer info"

b. [Java](#)

c. [Git](#) - It is recommended to install a [GUI Client](#), like [SmartGit](#).

There is also a command line tool [vcstool](#) that can perform batch Git operations on [groups of repositories](#).

d. [Gradle](#) - Follow the "Installation" and "Configuration" sections.

3. Choose and install an IDE:

Follow the "Installation" and "Configuration" sections in one of the following.

Make sure to setup the IHMC code formatter.

a. [IntelliJ](#)

More code navigation, refactoring, etc. tools

Gradle script editor

- Requires [JAR manifest workaround on Windows](#)

b. [Eclipse](#)

Faster compilation, quicker to launch simulations

Fantastic multi-monitor support

- Gradle files only editable via plain text

4. Use Git to create a repository-group w/ IHMC Open Robotics Software

- In your system home folder (or C:/ drive in Windows), clone a copy of [repository-group](#)

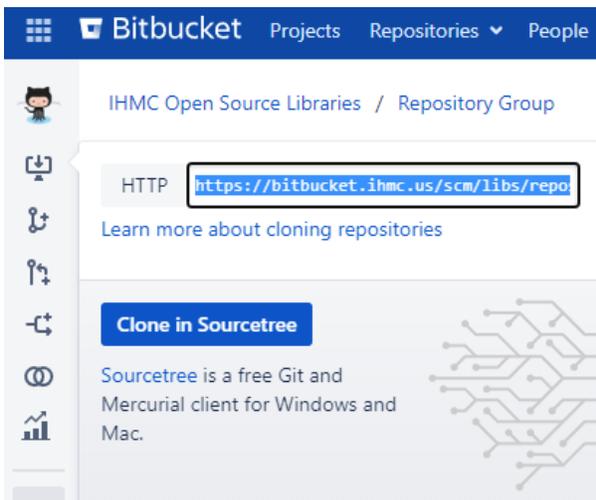
- Inside the [repository-group](#) folder, clone [ihmc-open-robotics-software](#)

It should look like:

```
repository-group
  ihmc-open-robotics-software
    build.gradle
    gradle.properties
    settings.gradle
  <other IHMC repositories as needed>
build.gradle
gradle.properties
settings.gradle
```

There are several choices for cloning a repository, including command line, graphical git tool, or your IDE. For example, in Eclipse, add the Git perspective and various Git tools will be available through the GUI, including clone.

To get the URL, navigate to the repository you wish to clone by searching <https://bitbucket.ihmc.us/>, then click on the clone button in the upper left. See official guide [here](#).



You may wish to clone more projects depending what you are working on. Atlas and Valkyrie simulations are included in Open Robotics Software but not their user interfaces. Take a look at what repositories are available here: [Repository Collections](#)

Note: If you clone any proprietary projects you will need to edit `~/ .gradle/gradle.properties` to contain the following:

```
artifactoryUsername=yourcrowdusername
artifactoryPassword=yourcrowdpassword
```

5. **Use your IDE to import the code you cloned**

When you import a Gradle project, the IDE is analyzing the Java build structure that Gradle comes up with and copying that configuration into the IDE's data structures as if you had painstakingly configured a new Java project manually using the IDE's UI.

Perform the Import directly on the `repository-group` folder or the `repository-group/build.gradle.kts` file.

Follow the "Importing Gradle Projects" section of your IDE's page.

- a. [IntelliJ IDEA#ImportingGradleProjects](#)
- b. [Eclipse IDE#ImportingGradleProjects](#)

6. **Setup network configuration**

- a. [Reserve an IP address](#) if necessary.
- b. Reserve a [DDS domain ID](#).
- c. Run `NetworkParametersCreator.java` and save it.
- d. Don't forget to set the `RTPSSubnet`. (i.e. `RTPSSubnet=10.100.1.34/24`)

7. **To see if everything is working, run some sample simulations.** For example `SpringFlamingoSimulation.java` or `AtlasObstacleCourseDemo.java`

It's recommended to skim over the guides and try to understand a bit about how things are working:

- Software builds
 - [IHMC Build](#)
 - [Repository Group](#)
- Operating systems
 - [Ubuntu](#)
 - [Windows](#)
 - [MacOS](#)
- Version control
 - [Git](#)
 - [vcstool](#) - Batch Git operations on groups of repos on command line
 - [SmartGit \(install guide, license files\)](#) - Requires license, most feature complete Git GUI
 - [IntelliJ VCS Plugin](#) - Fancy, nice to use, but can be sluggish
 - [Eclipse EGit VCS Plugin](#) - Not many features, buggy
- IDEs
 - [IntelliJ IDEA](#)
 - [Eclipse IDE](#)
- [Java](#)
- [Gradle](#)

Software Development

Okay, now you're ready to write code.

Check out the [IHMC Employee Handbook](#).

- [IHMC Java Code Style and Conventions](#)
- [JavaDoc Style Guide](#)
- [IHMC Git Handbook](#)

- [Creating a Pull Request](#)
- [Reviewing a Pull Request](#)

General Troubleshooting

System and application crashes - GC Overhead Exceptions

Sometimes applications or your entire system crashes. This can be due to being on a less capable machine or just running resource expensive or buggy applications.

1. Reboot your machine
2. Close out as many background applications as possible after booting
3. Launch a system monitor where you can see CPU and RAM usage. Preferably a graph over time so you don't miss something.
4. Launch IntelliJ and keep an eye on CPU and RAM
5. Proceed slowly and wait for CPU to go to <5% stably before clicking the next thing