

The Java archive file, `AardMessage.jar`, includes the required ZeroMQ libraries and messaging client `AardMsg`. This client class will allow you to communicate with the navigation server.

- In the folder that contains your java solution, create a subfolder named `lib`, and save `AardMessage.jar` to this folder.
- If you are using Eclipse
  - Right-click the project in the Package Explorer, then select Properties.
  - In the Libraries tab select “Add External Jars...”, then select `AardMessage.jar` from the `lib` folder. When you run the program, it will automatically include the jar file.
- When compiling and running your program, include the jar file in the classpath. Most IDEs will have the ability to include libraries.

When using Windows CMD: Navigate the directory where your program is and execute the commands below to compile and run. (If you are using a different OS you may have to change the syntax slightly, see Mac below)

```
> javac -cp lib\AardMessage.jar <TheNameOfMyFormationClient>.java
> java -cp ;lib\AardMessage.jar <TheNameOfMyFormationClient>
```

Otherwise when using Mac CMD: Navigate the directory where your program is and execute the commands below to compile and run. (If you are using a different OS you may have to change the syntax slightly)

```
> javac -cp ./lib/AardMessage.jar <TheNameOfMyFormationClient>.java
> java -cp ../lib/AardMessage.jar <TheNameOfMyFormationClient>
```

#### Usage:

You can create an instance of the `AardMsg` class in your program by specifying the Formation server hostname or IP address in the constructor, or by using the default constructor, which will use `localhost`. Once instantiated, `AardMsg` can connect to the Formation server by calling `connect()`. Once connected to the server, use the `send(String message)` and `receive()` methods to communicate. When the challenge has completed, use `disconnect()` to close the connection to the server.

Example Implementation:

```
public class AardFormationClient {
```

```
public static void main(String[] args) throws Exception {  
    AardMsg msg = new AardMsg(); //By default will connect to localhost  
    msg.connect();  
  
    msg.send("thrust +x");  
    msg.receive();  
  
    //  
    // Your code here  
    //  
  
    msg.disconnect();  
}  
}
```