

<https://discordapp.com/channels/187635036525166592/310517733458706442/720620305944412220/>

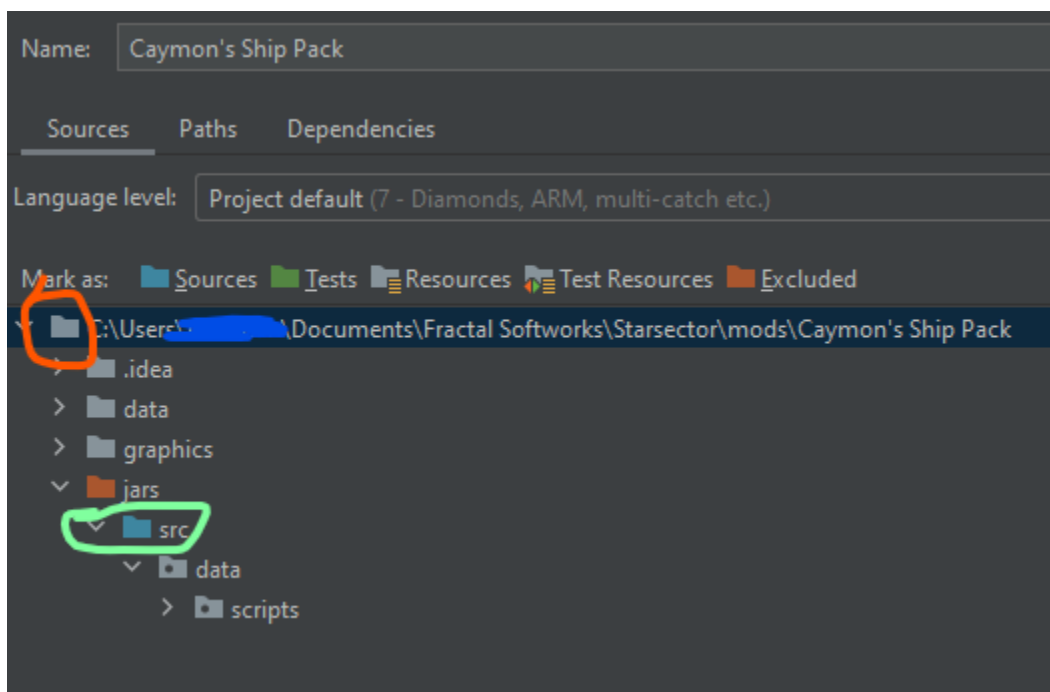
IntelliJ and you: How to finish setting it up

Tomatopaste already basically did a step by step guide walking asta through how to set it up properly

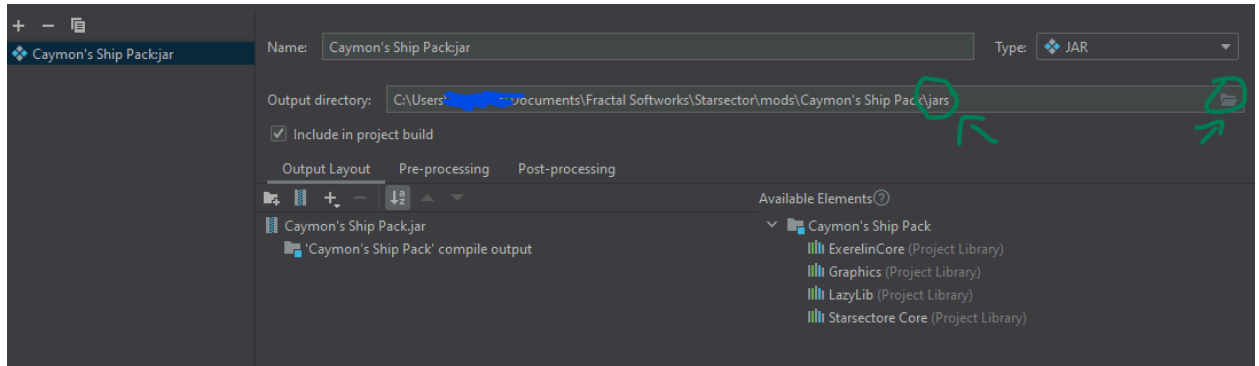
(Note: When i say guide, I meant he walked asta through the process of setting up intelliJ. If you already did the basics of setting up intelliJ, then this is basically just filling in the blanks on what to do to finish setting up so you too don't spend like all afternoon trying to figure out why your customstart isn't working and it ends up because you never told how to set it up to compile correctly and read your mod's jar.)

This is additional instructions to the forum's instructions on setting up intelliJ

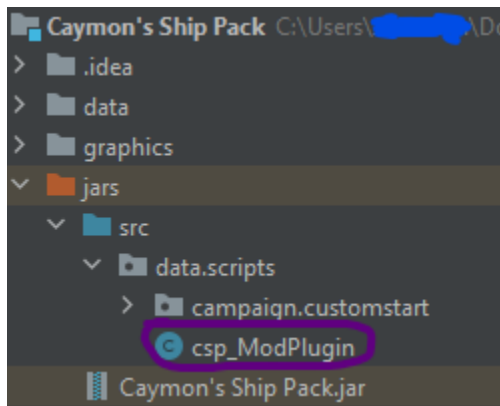
(<https://fractalsoftworks.com/forum/index.php?topic=10057.0>) as the forum leaves out a lot of information that as it turns out, you kinda need if you wanna add hullmods and shit. One way to do this is having it automatically compile directly into your mod directory.



You see this? If you followed the forum's instructions, the source folder would be the orange circle. What you want to do is make 2 folders called jars and a folder called src inside jars and make src the source folder (See Light Green circle)



You want to change the output directory to the jars file in the artifacts section in project structure.



You need a Modplugin so the game can load your Jar. Make a new folder called data inside src and another folder called scripts inside data. Then make a Java class and name it "X_ModPlugin" (With X obviously a stand in for your mod's name), It should be inside data/scripts (Don't worry, data.scripts is normal, you didn't do anything wrong).

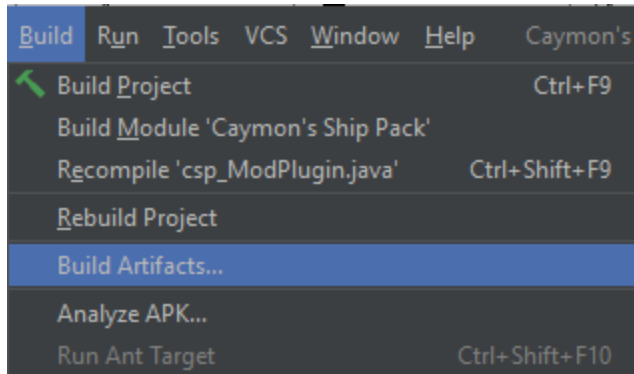
- Now inside the modplugin, copy and paste the following:

```
package data.scripts;
import com.fs.starfarer.api.BaseModPlugin;
import com.fs.starfarer.api.Global;

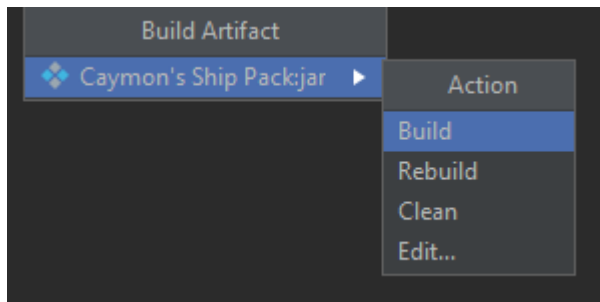
public class csp_ModPlugin extends BaseModPlugin {

}
```

(Note: Change "csp_ModPlugin" to whatever you named the mod plugin so if you named it like Ironshell_ModPlugin, it should be "Public class Ironshell_ModPlugin extends BaseModPlugin")



- Now go to build near the top left and click on it and go to “Build Artifacts...”, click on that



- Click on Rebuild and it will rebuild it and compile it into a jar that will show up in jars if you did everything right.

Project name:

Caymon's Ship Pack

Project SDK:

This SDK is default for all project modules.

A module specific SDK can be configured for each of the modules as required.

1.7 version 1.7

Edit

Project language level:

This language level is default for all project modules.

A module specific language level can be configured for each of the modules as required.

7 - Diamonds, ARM, multi-catch etc.

Project compiler output:

This path is used to store all project compilation results.

A directory corresponding to each module is created under this path.

This directory contains two subdirectories: Production and Test for production code and test sources, respectively.

A module specific compiler output path can be configured for each of the modules as required.

C:\Users\... \Documents\Fractal Softworks\Starsector\mods\Caymon's Ship Pack\jars

- (Optional prob): Change the project output to the jars file also. I don't think this is super important to do but i don't think there's any harm in doing so. Someone smarter than me will correct me if I'm wrong. I'm sure.

```

    "id": "Csp",
    "name": "Caymon's Ship pack",
    "author": "Caymon",
    "version": "1.0.4.1",
    "description": "Just another Ship pack.",
    "gameVersion": "0.9.1a",
    "jars": ["jars/Caymon's Ship Pack.jar"],
    "modPlugin": "data.scripts.csp_ModPlugin",

```

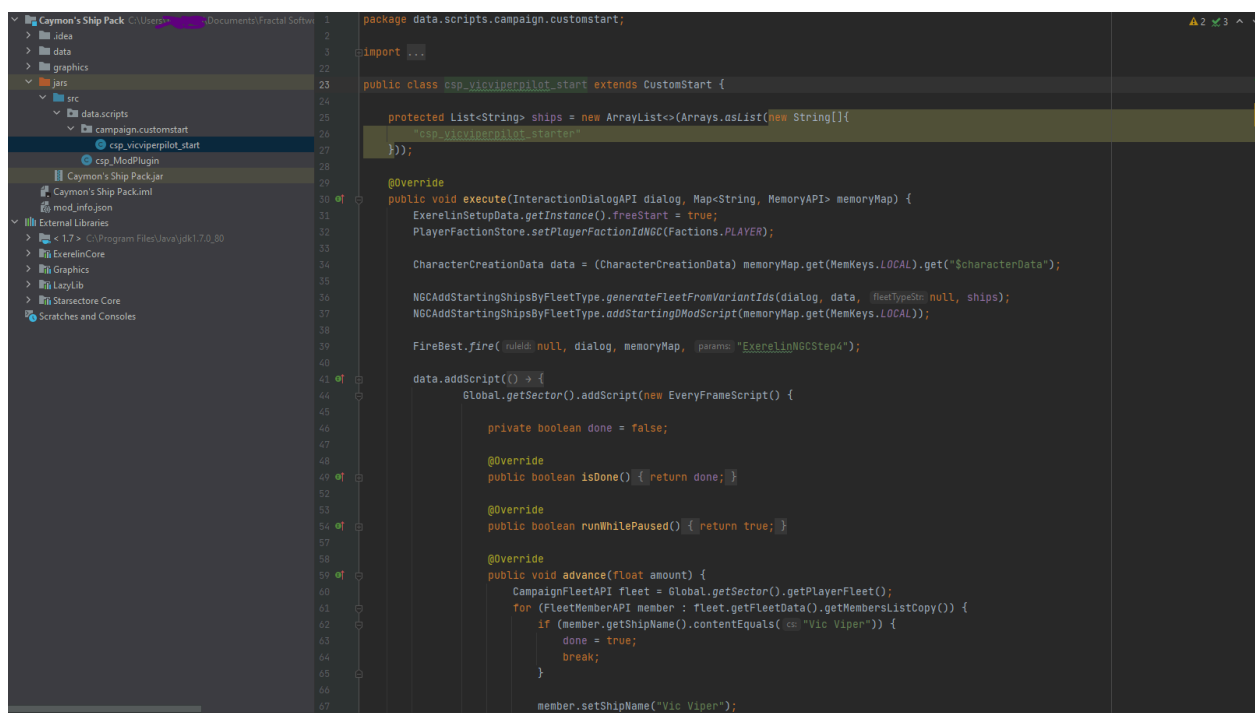
- - Go to your mod's modinfo.json file and add:
 - "jars":["jars/x.jar"] (Replace x with your mod's name here obv)
 - "modPlugin":"data.scripts.x_ModPlugin",
- Now that you have done all of that, you should be fine and able to compile shit into a jar for your mod and have it work in game prob. Put any code that cant be compiled by the game itself inside src+ folders. Ex: Customstarts,hullmods,shipsystems, star systems prob, etc

How to make a Custom start with Nex

Documents > Fractal Softworks > Starsector > mods > Caymon's Ship Pack > data > config > exerelin

Name	Date modified	Type	Size
 customStarts	3/16/2021 3:41 PM	JSON File	1 KB

- First make a config/exerelin folder in your mod, then you wanna get a customstarts.JSON file, you can grab one from either my mod or a mod like interstellar imperium with a customstart of its own.



```
package data.scripts.campaign.customstart;

import ...

public class csp_vicvipilot_start extends CustomStart {

    protected List<String> ships = new ArrayList<>(Arrays.asList(new String[]{
        "csp_vicvipilot_start"
    }));

    @Override
    public void execute(InteractionDialogAPI dialog, Map<String, MemoryAPI> memoryMap) {
        ExerelinSetupData.getInstance().freeStart = true;
        PlayerFactionStore.setPlayerFactionID(NGC(Factions.PLAYER));

        CharacterCreationData data = (CharacterCreationData) memoryMap.get(MemKeys.LOCAL).get("$characterData");
        NGCAddStartingShipsByFleetType.generateFleetFromVariantIds(dialog, data, fleetType: null, ships);
        NGCAddStartingShipsByFleetType.addStartingDModScript(memoryMap.get(MemKeys.LOCAL));

        FireBest.fire(null, dialog, memoryMap, param: "ExerelinNGCStep4");

        data.addScript(() -> {
            Global.getSector().addScript(new EveryFrameScript() {

                private boolean done = false;

                @Override
                public boolean isDone() {return done;}

                @Override
                public boolean runWhilePaused() {return true;}

                @Override
                public void advance(float amount) {
                    CampaignFleetAPI fleet = Global.getSector().getPlayerFleet();
                    for (FleetMemberAPI member : fleet.getFleetData().getMembersListCopy()) {
                        if (member.getShipName().contentEquals("Vic Viper")) {
                            done = true;
                            break;
                        }
                    }
                    member.setShipName("Vic Viper");
                }
            });
        });
    }
}
```

- Then what you wanna do is make a customstart java file. If you wanna go the way of 1 single ship for a custom start, get the java file from my mod or the imperium's yamato start java file and change it to fit your mod's needs. Note that if you use imperium's yamato start as a base, remember to remove all the things referencing swp. Note: the ship referenced in the java file should be a variant.
 - Also make sure the java file is located within the src folders so it can be compiled into your mod's jar. You will need to make a scripts,campaign and customstart folder for it.

```

{
  "starts": [
    {
      "id": "csp vicviperpilot_start",
      "name": "Superfighter Vic Viper",
      "difficulty": "Normal",
      "desc": "Pilot this experimental fighter and shoot up your enemies!",
      "className": "data.scripts.campaign.customstart.csp_vicviperpilot_start",
      "factionId": "player",
      "requiredModId": "Csp",
      "randomSector": 0,
    },
  ],
}

```

- When setting up your custom start file, make sure the id matches the circled part of the classname. (Which is also the name of your java file). ModId is the id in your modinfo.json (It is cap sensitive) Everything else should be self explanatory.

Debugging Guide from Wisp:

1. From the toolbar, select the dropdown and then "Edit Configurations..."

2. Edit the values in the window to match those shown here.

3. On the toolbar, click the Debug icon to launch the game and automatically attach the debugger.

- 1. Go to the toolbar and select dropdown and then click on "Edit Configurations"
- 2. Edit the values so they match those to what's in the picture, this changes the vram you can change it to match that for your standard modlist and your own vram

- 3. Go to modify options and change your options so it matches that to those in the picture.
- 4. Go the toolbar and click debug to launch the game and attach the debugger.