

Tutorial 1 — First Co-Simulation

Overview

This INTO-CPS tutorial will show you how to:

1. Set up a co-simulation project in Visual Studio Code
2. Run a co-simulation from within Visual Studio Code

Requirements

The tutorial assumes that you have the following pieces of software installed:

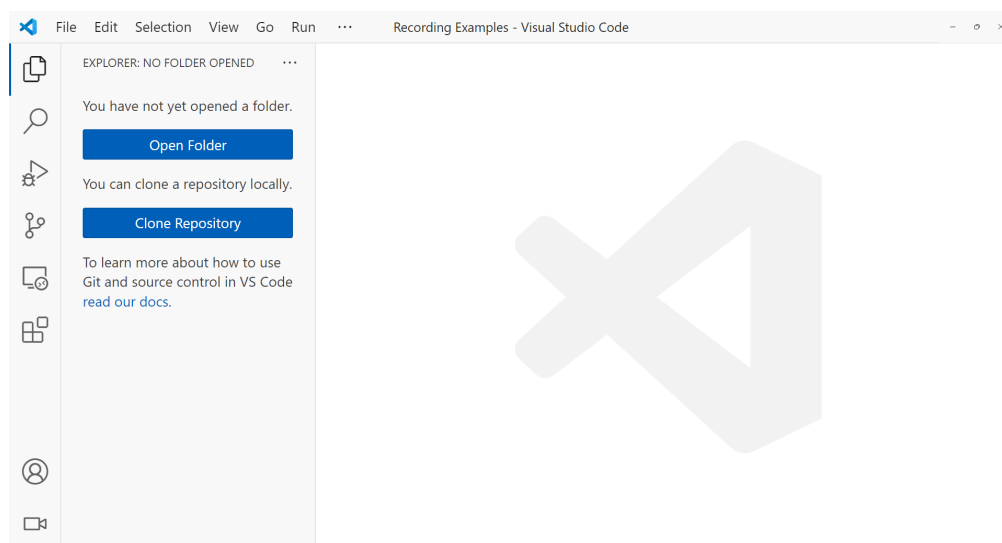
- Visual Studio Code
- `Cosimulation Studio` extension for Visual Studio Code
- Java SE Runtime Environment 11, newer versions may also work
- `Maestro COE` (Co-simulation Orchestration Engine) w. Web API

Follow Tutorial 0 if any of these are missing from your system.

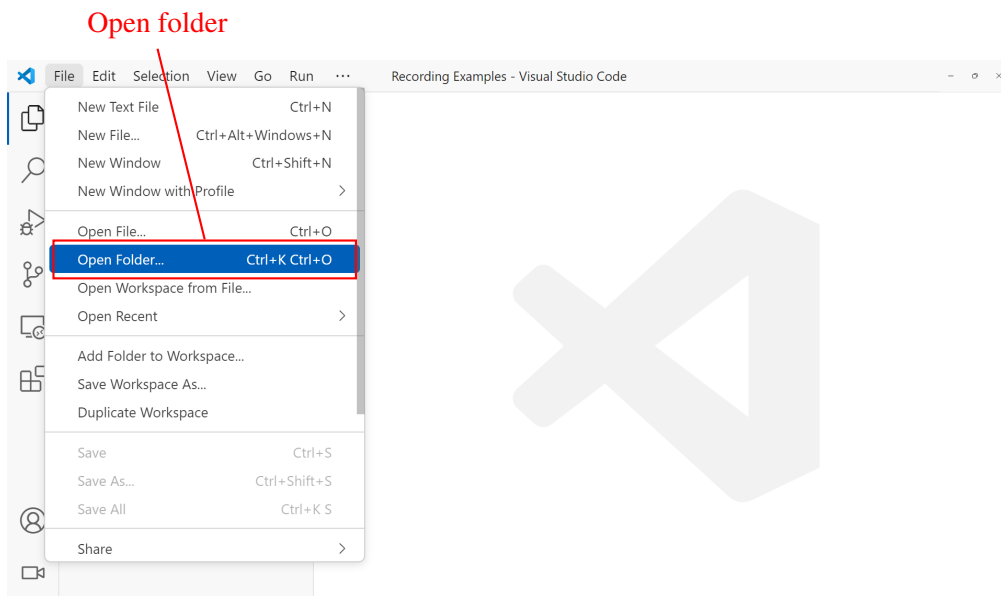
1 Set up a co-simulation project in Visual Studio Code

Any workspace folder containing a `cosim.json` simulation configuration file is automatically treated as a co-simulation project, enabling features such as autocompletion, linting and integration with the `Maestro COE`. Setting up a project is thus as simple as opening a folder in Visual Studio Code and writing a `cosim.json` file. This section of the tutorial describes this process.

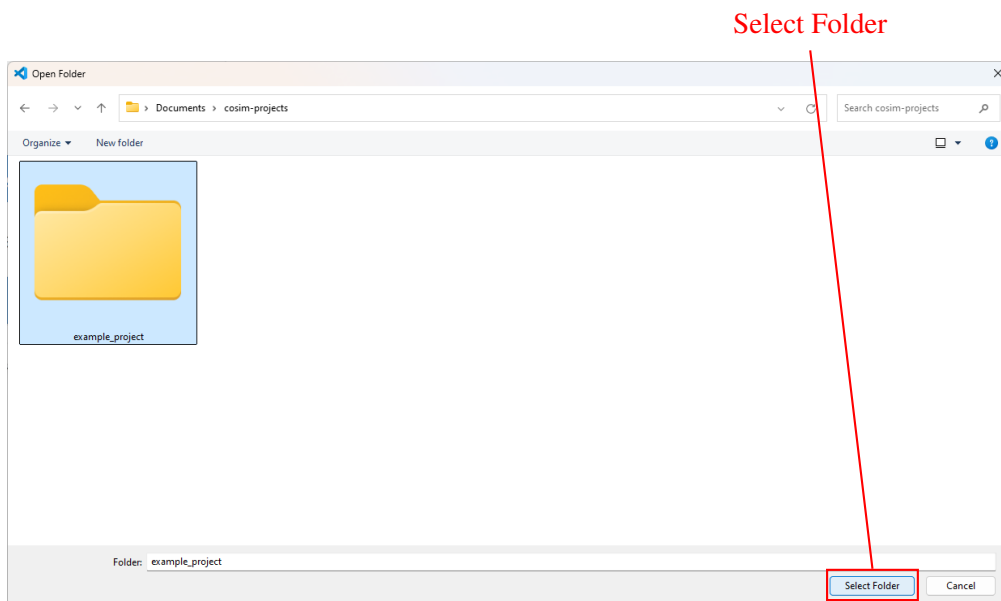
Step 1. Launch Visual Studio Code. It is not important if the view is empty as the one below or if Visual Studio Code opens up an existing project.



Step 2. To open the folder that will contain the project, select *File > Open folder...*

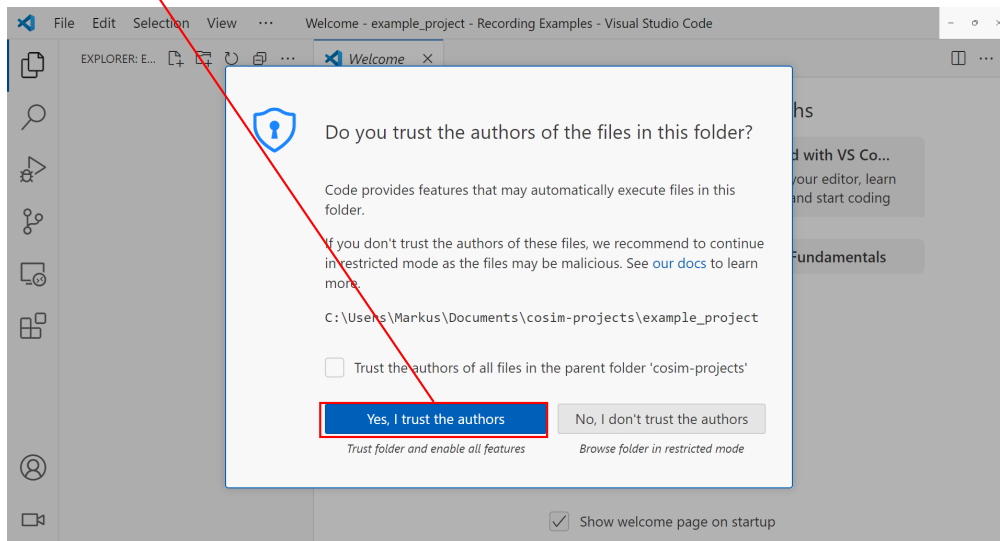


Step 3. Select an empty folder or create a new one.



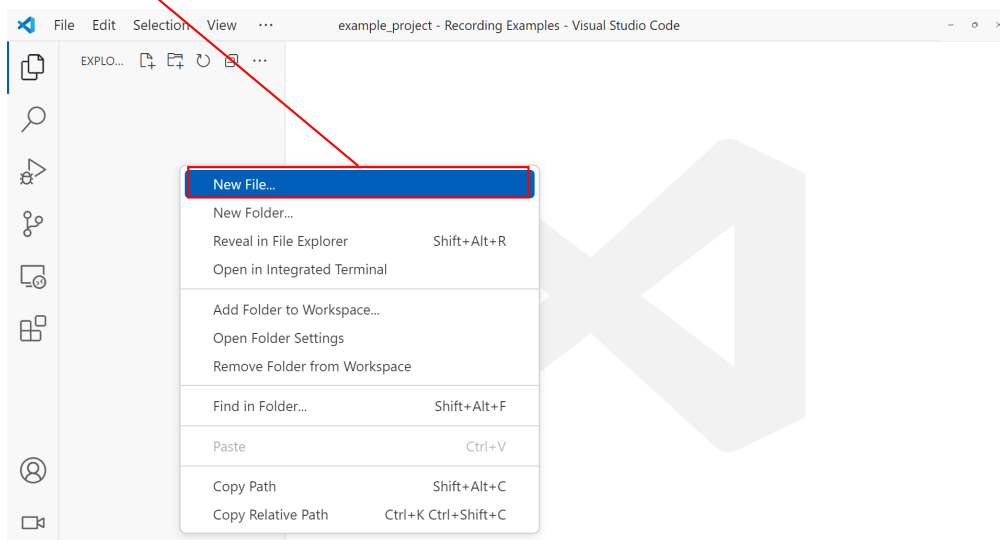
Step 4. A pop-up will appear, asking you to confirm that you trust the authors of the files in the selected folder. You need to confirm for the language features of the extension to work.

Trust the authors



Step 5. Create a new file named `cosim.json`.

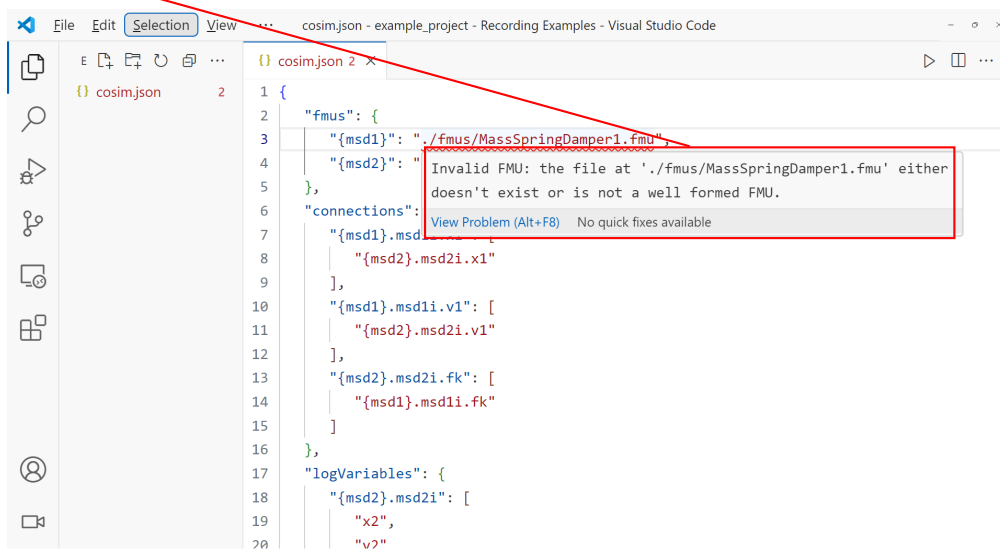
New File...



Step 6. We will be using the example found in the tutorial archive. Copy-and-paste the contents of the example `cosim.json` file into your own `cosim.json`.

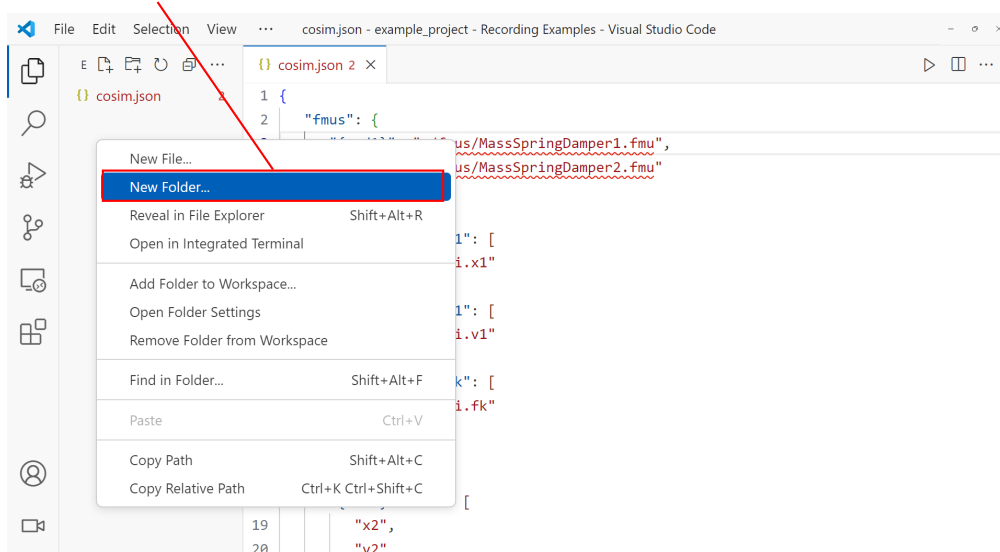
Step 7. The paths referring to the location of the FMUs should now be underlined with a red squiggle. If you hover over the path, you should see that the reason for the error is that the extension cannot find a valid FMU at that location.

Error



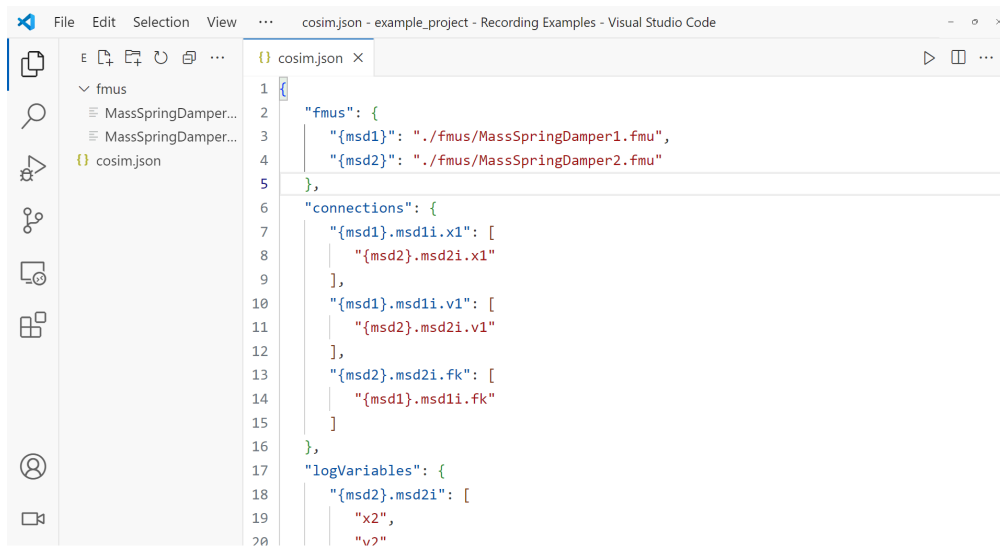
Step 8. To make the FMUs available to the extension, we will create a new folder named `fmus` in the workspace to contain the FMUs.

New Folder...



Note: The extension does not impose any restrictions on the structure of the project, so you can name the folder anything you like, remembering to update the corresponding path in `cosim.json`.

Step 9. Populate the newly created folder with the two FMUs found in the directory alongside this tutorial PDF. The errors indicating that the FMUs were not found should now have cleared. At this point, we have a very basic project set up that is ready for simulation.



The screenshot shows the Visual Studio Code editor with a file named `cosim.json` open. The editor has a sidebar on the left with a file explorer showing a folder `fms` containing two files: `MassSpringDamper1.fmu` and `MassSpringDamper2.fmu`. The main editor area displays the JSON content of `cosim.json` with line numbers 1 through 20. The JSON defines two FMU instances, `msd1` and `msd2`, and their connections. `msd1` is connected to `msd2` via `msd1i.x1` and `msd2i.x1`. `msd1` also has a connection to `msd2` via `msd1i.v1` and `msd2i.v1`. The `logVariables` section lists `x2` and `v2` for `msd2`.

```
1 {
2   "fms": {
3     "{msd1}": "../fms/MassSpringDamper1.fmu",
4     "{msd2}": "../fms/MassSpringDamper2.fmu"
5   },
6   "connections": {
7     "{msd1}.msd1i.x1": [
8       "{msd2}.msd2i.x1"
9     ],
10    "{msd1}.msd1i.v1": [
11      "{msd2}.msd2i.v1"
12    ],
13    "{msd2}.msd2i.fk": [
14      "{msd1}.msd1i.fk"
15    ]
16  },
17  "logVariables": {
18    "{msd2}.msd2i": [
19      "x2",
20      "v2"
21    ]
22  }
23 }
```

2 Run a co-simulation from within Visual Studio Code

Step 10. Before running a simulation, ensure that **Maestro** is running on your local device. Open a terminal in the folder where you have installed **Maestro**, and then run

```
java -jar maestro-webapi-<latest-version>-bundle.jar.
```

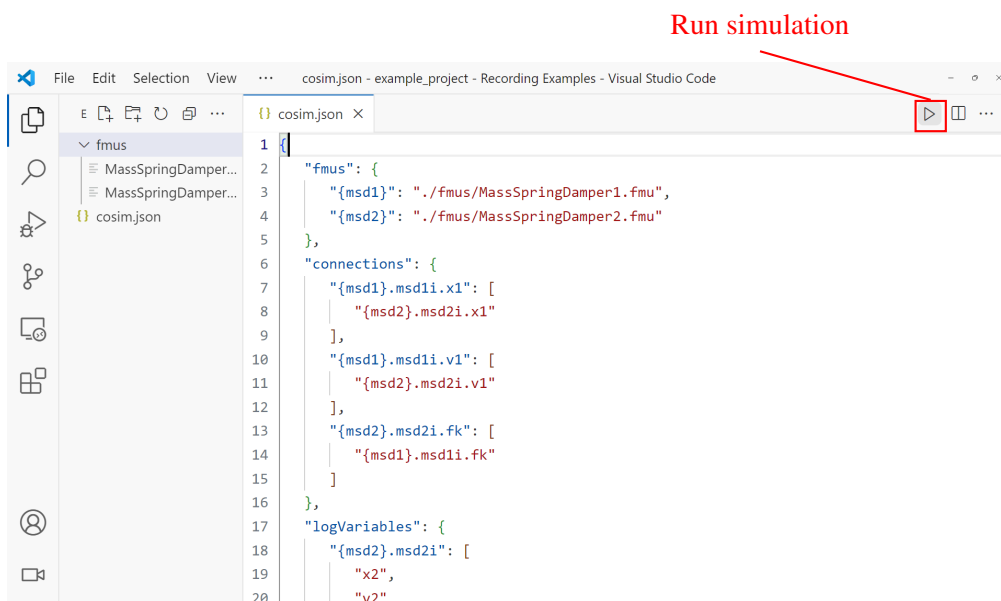
You should see something similar to the image below:

```
markus:-$ java -jar maestro-webapi-2.4.1-bundle.jar
Version: 2.4.1

  ____ _
 / ___ \ | |
| |___ \| |_| | |
| |___) | | | |
|_____)_|_|_|_|
:: Spring Boot ::      (v2.2.7.RELEASE)

19:19:00.401 [background-preinit] WARN org.springframework.http.converter.json.Jackson2ObjectMapperBuilder - For Jackson Kotlin classes support please add "com.fasterxml.jackson.module:jackson-module-kotlin" to the classpath
19:19:01.221 [main] INFO org.apache.coyote.http11.Http11NioProtocol - Initializing ProtocolHandler ["http-nio-8082"]
19:19:01.222 [main] INFO org.apache.catalina.core.StandardService - Starting service [Tomcat]
19:19:01.222 [main] INFO org.apache.catalina.core.StandardEngine - Starting Servlet engine: [Apache Tomcat/9.0.34]
19:19:01.269 [main] INFO org.apache.catalina.core.ContainerBase.[Tomcat].[localhost].[/] - Initializing Spring embedd
ed WebApplicationContext
19:19:02.141 [main] INFO org.apache.coyote.http11.Http11NioProtocol - Starting ProtocolHandler ["http-nio-8082"]
```

Step 11. Run the simulation from Visual Studio Code.



Step 12. After running for a short period, the results show up in a new file

```
File Edit Selection • time,[msd2].msd2i.fk,[msd2].msd2i.v2,[ms • Untitled-1 - example_project - Recording Examples - Vi...
cosim.json
  fmus
    MassSpringDamper...
    MassSpringDamper...
  cosim.json
1 time,{msd2}.msd2i.fk,{msd2}.msd2i.v2,{msd2}.msd2i.x2,{msd1}.msd1i.v1,{msd1}.msd1i.x1
2 0.0,-2.0,0.0,0.0,0.0,1.0
3 0.001,-1.99980067,0.018993375,9.9665E-5,-0.02655000416666664,0.9998366666666666
4 0.002,-1.99980067,0.018993375,9.9665E-5,-0.02655000416666664,0.9998366666666666
5 0.003,-1.99980067,0.018993375,9.9665E-5,-0.02655000416666664,0.9998366666666666
6 0.004,-1.99980067,0.018993375,9.9665E-5,-0.02655000416666664,0.9998366666666666
7 0.005,-1.99980067,0.018993375,9.9665E-5,-0.02655000416666664,0.9998366666666666
8 0.006,-1.99980067,0.018993375,9.9665E-5,-0.02655000416666664,0.9998366666666666
9 0.007,-1.99980067,0.018993375,9.9665E-5,-0.02655000416666664,0.9998366666666666
10 0.008,-1.99980067,0.018993375,9.9665E-5,-0.02655000416666664,0.9998366666666666
11 0.009000000000000001,-1.99980067,0.018993375,9.9665E-5,-0.02655000416666664,0.9998366666666666
12 0.010000000000000002,-1.9989727349720643,0.039592802286559176,3.9729918063444896E-4,-e
13 0.011000000000000003,-1.9989727349720643,0.039592802286559176,3.9729918063444896E-4,-e
14 0.012000000000000004,-1.9989727349720643,0.039592802286559176,3.9729918063444896E-4,-e
15 0.013000000000000005,-1.9989727349720643,0.039592802286559176,3.9729918063444896E-4,-e
16 0.014000000000000006,-1.9989727349720643,0.039592802286559176,3.9729918063444896E-4,-e
17 0.015000000000000007,-1.9989727349720643,0.039592802286559176,3.9729918063444896E-4,-e
18 0.016000000000000008,-1.9989727349720643,0.039592802286559176,3.9729918063444896E-4,-e
19 0.017000000000000009,-1.9989727349720643,0.039592802286559176,3.9729918063444896E-4,-e
20 0.01800000000000001,-1.9989727349720643,0.039592802286559176,3.9729918063444896E-4,-e
```

Congratulations!

You have now successfully set up and simulated your first co-simulation project in Visual Studio Code.