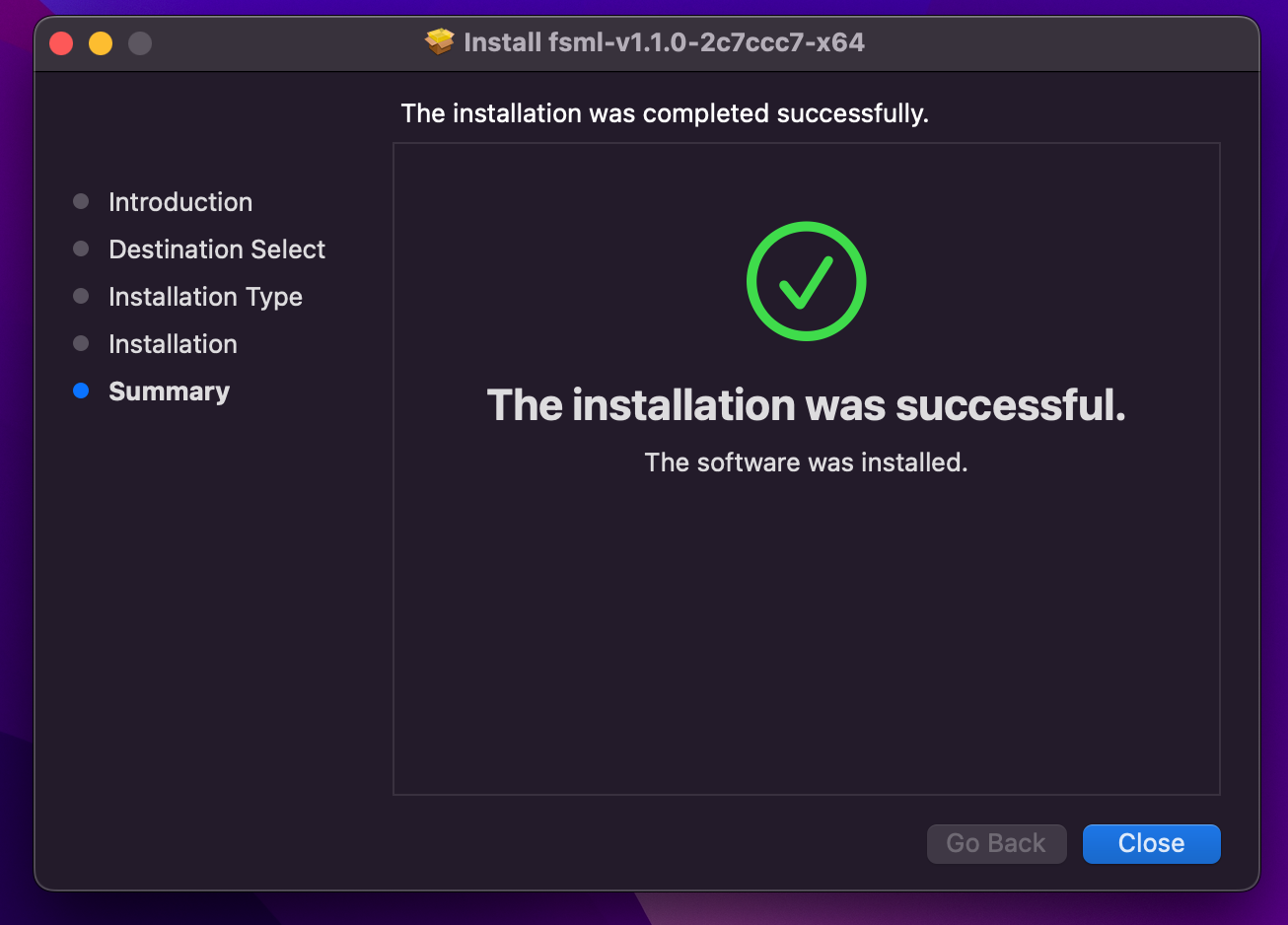
Instruction steps to run the FSML CLI tool and generate an FSML manifest out of the example MicroByre Lector Data files.

### 1. Download the following files (those in this folder):

* the **microbyreParser.ts** (this is the plugin parser for the lecto data files).
* one of the data files (here we will use **microbyreData.csv**)
* the **fsml-v1.1.0-2c7ccc7-x64.pkg** FSML CLI installer (this requires a MacOS machine)

### 2. Installing the FSML CLI Tool

Double-click on the installer file and follow through the MacOS installer steps.



Note that if you are in a debian machine you can download the proper installers from <https://github.com/TeselaGen/fsml.org/releases/tag/v.1.1.0>

### 

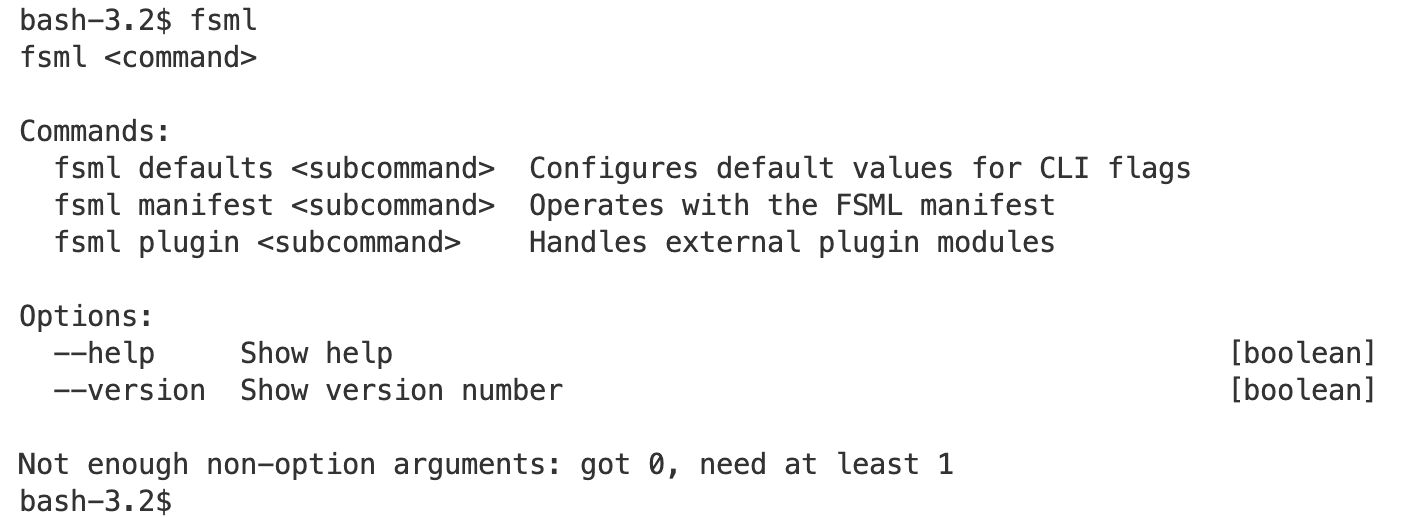
### 3. Trying out the FSML CLI Tool

Open a terminal window and type the following command

*(note that the “****$>****” symbol is not to be typed, it's just used here to represent your CLI prompt)*

**$> fsml**

You should see the FSML CLI helper docs.



You can navigate into the CLI docs by typing the different commands (e.g., “$> **fsml defaults**”).

### 4. Installing the Microbyre plugin

We need to install the **microbyreParser.ts** plugin that contains the custom logic to parse the Microbyre Data Lector files.

To do so, run the following commands:

**$> fsml plugin install microbyreParser --from-url file://$(pwd)/microbyreParser.ts**

Note that the ‘$(pwd)’ part of the --from-url option is just to get you working directory, this assumes you have your parser located here.

Now to double-check the parser is properly installed you can run the following command:

**$> fsml plugin list**

You should see something like this



### 5. Parsing Data Files into an FSML Manifest

Now that we have the proper parser installed we can use it to parse one of the Data Lector Files. Run the following command:

**$> fsml manifest create microbyreData.csv --parser microbyreParser**

**--type data --author** YOUR\_NAME **--format json --write mbManifest --pack tar**

Noted all the flags passed to the command? Well, we can set some of them as defaults so we don’t have to pass them in every time.

To do so we can use the “defaults” command.

**$> fsml defaults set manifest.type data**

**$> fsml defaults set manifest.author** YOUR\_NAME

**$> fsml defaults set manifest.format json**

**$> fsml defaults set manifest.write myManifest**

**$> fsml defaults set manifest.pack tar**

Now you can try the previous command with fewer flags.

**$> fsml manifest create microbyreData.csv --parser microbyreParser**

That’s it, you should have generated some **.tar** files. You can untar them and you shall see your fsml.json manifest file.

Feel free to continue exploring the different commands and implementing your own plugin parsers following the Plugin Framework.