

WORKSHOP

Turbocharging collaboration

Contents

Introduction	1
Workspace export.....	1
Version control.....	5
Setup.....	6
Modifying code	10

Introduction

This workshop will highlight the collaborative concepts of workspace sharing and version control of you modelica code.

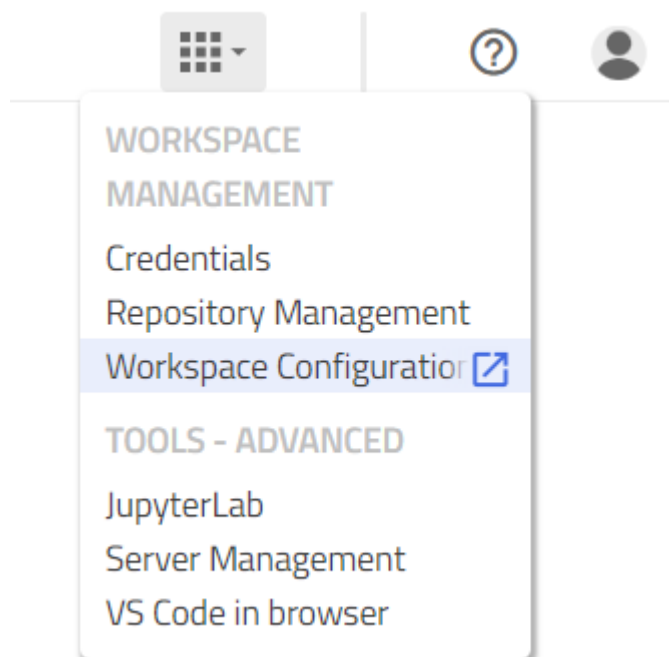
Workspace export

In the following workshop, you will

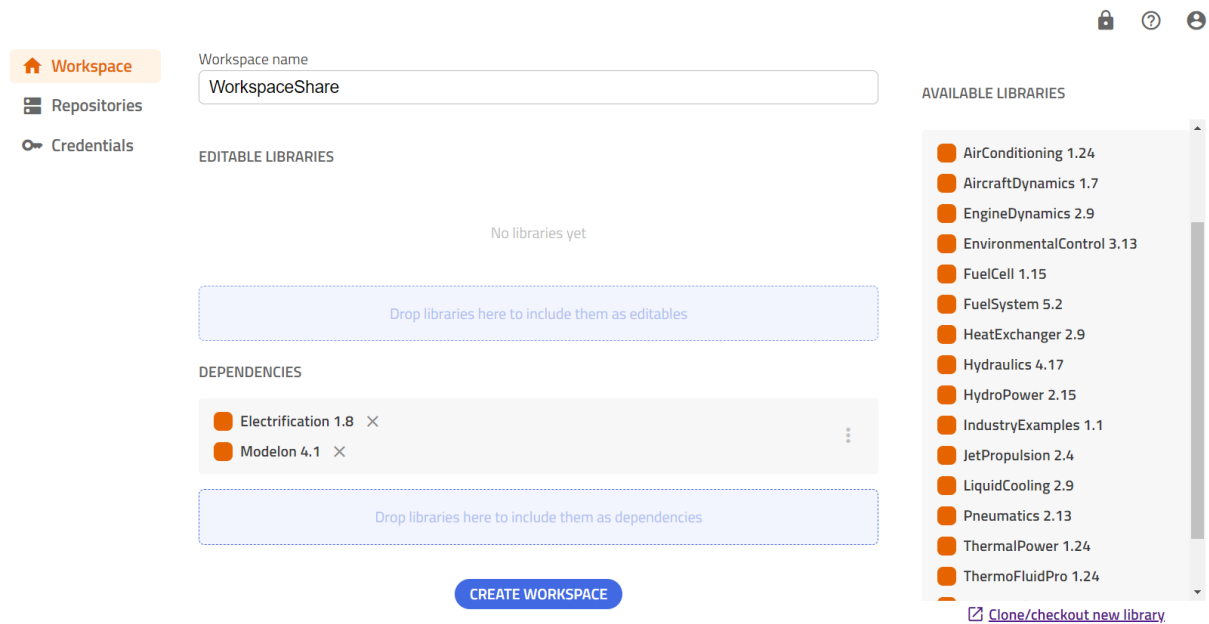
- Configure a workspace
- Duplicate a model into the Workspace package
- Simulate the model, to create a result file
- Add a plot view
- Export the Workspace.

Start by going to the Modelon Impact Landing page.

1. Open the Workspace configurator App in the top right corner:



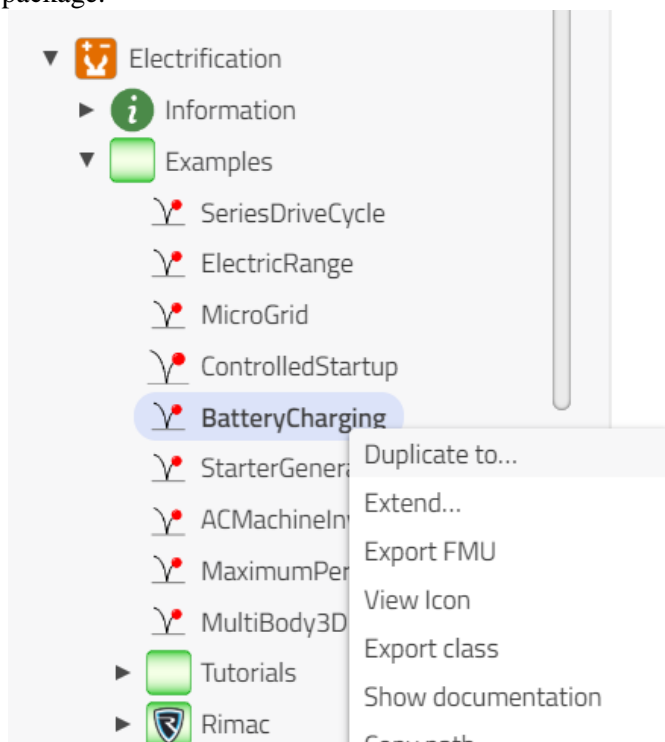
2. Create a new workspace according to the following specifications:

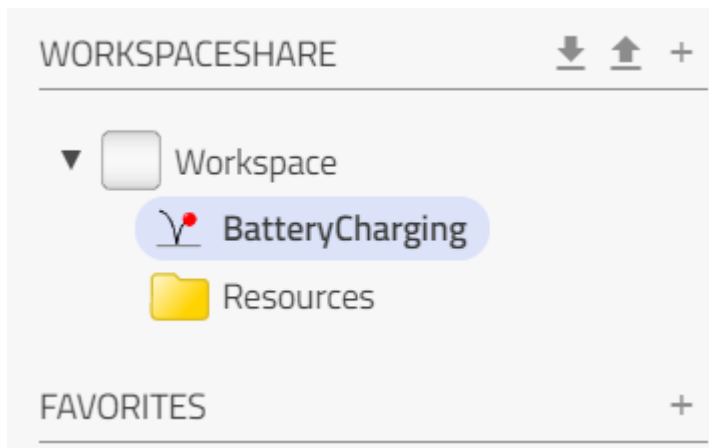


3. Press “Create Workspace”
4. Click the link in the bottom of the page, to open the workspace in Modelon Impact:

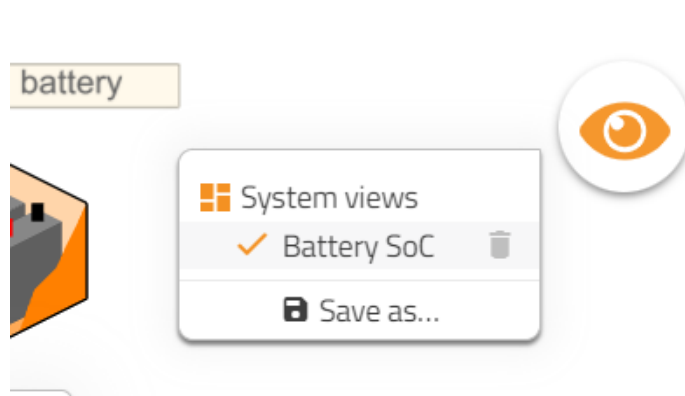
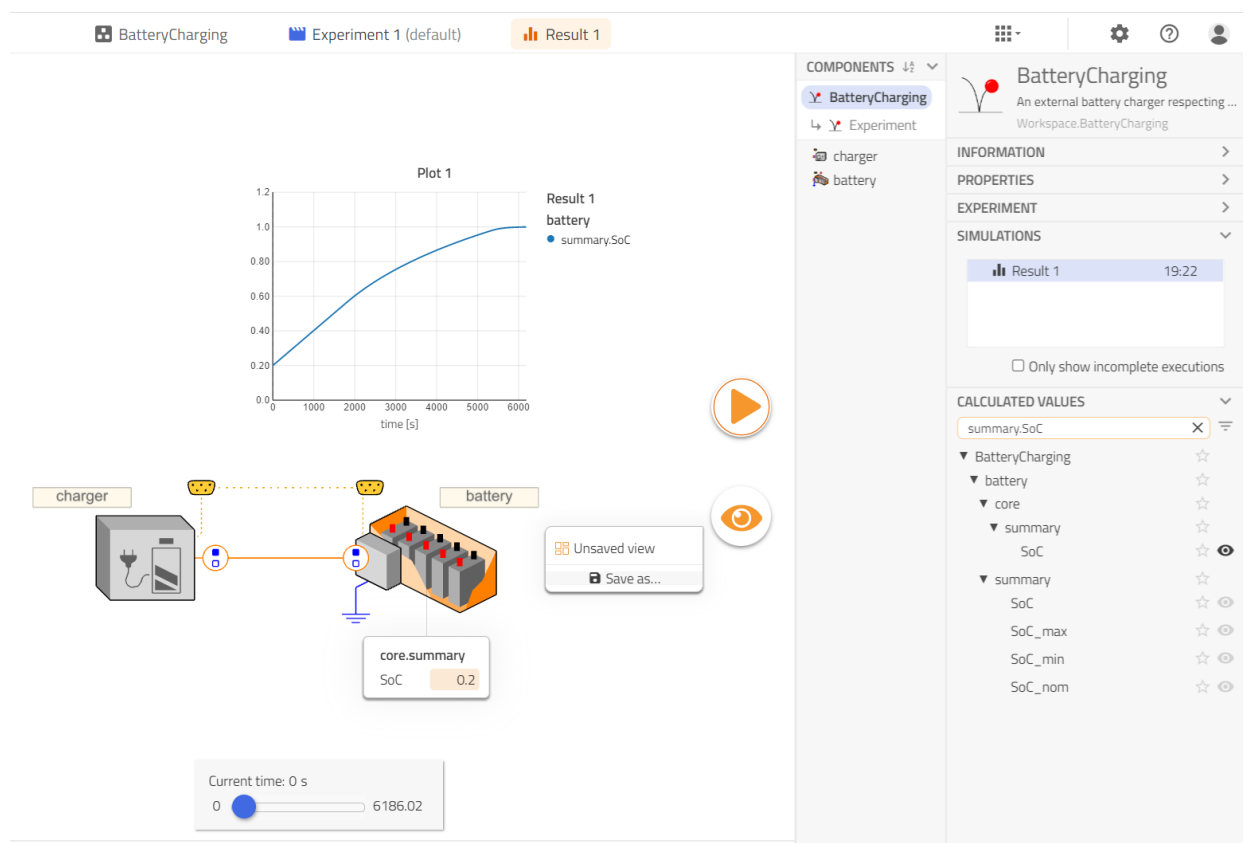
Open in [Modelon Impact](#) or [VS Code](#)

5. Go to the package browser and find the model “Electrification.Examples.BatteryCharging”
6. Right click the class and choose “Duplicate to...” and create a copy on the Workspace package.





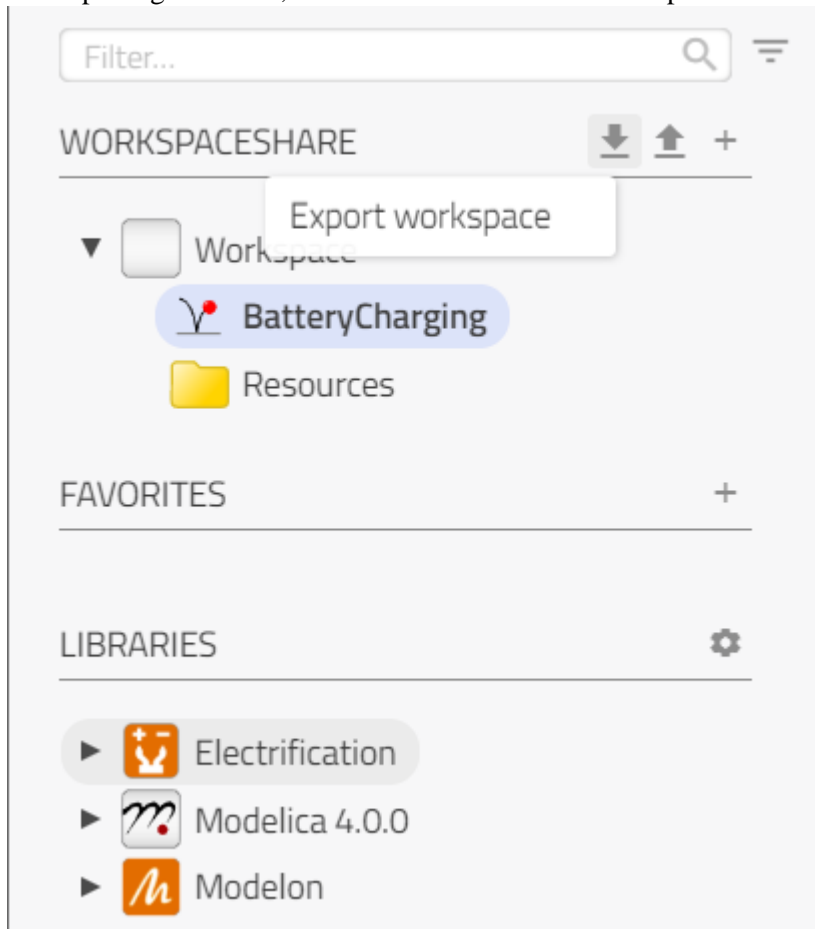
7. Open the model “Workspace.BatteryCharging” and simulate the model.
8. In the result, search for “summary.SoC”.
9. Plot the variable by dragging it into the canvas, then click the “eye” to add a stickie.
10. Save this view, Save as “Battery SoC”:



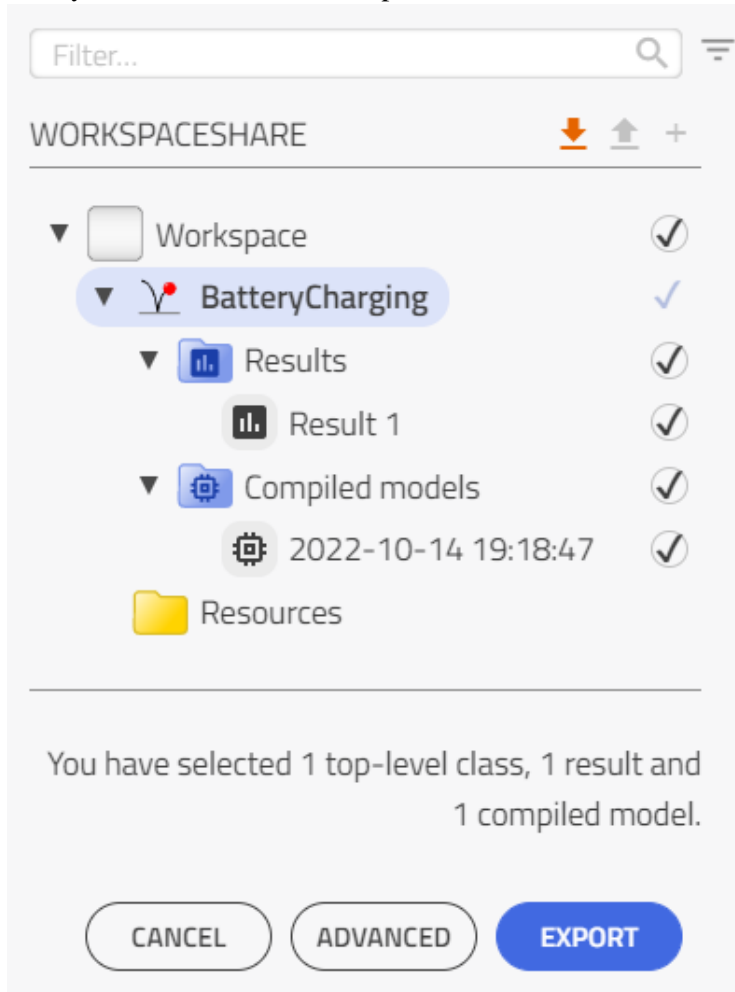
You can now activate or deactivate the view by clicking it again.

If you would like to share your work with a colleague, including the results and plots, you can export the whole workspace.

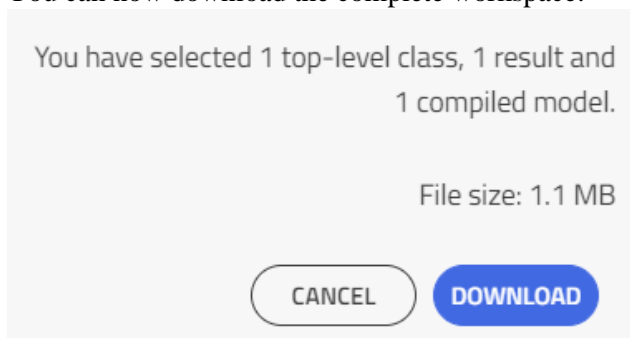
1. In the package browser, click the downward arrow to export:



- Now you can choose what to export:



- Press “Export” to create a .zip file.
- You can now download the complete workspace:



When you import a workspace, the name will be retrieved from the .zip file name. If you want to import it with a different name, you can rename the .zip file.

Optional: Open the .zip file to inspect its content.

Version control

In the following workshop you will investigate the possibility to work with a library imported from the code version management system GitHub.

As a reference repository, we will use the Modelon Training material, that you can find using the following link:

<https://github.com/modelon/Training>

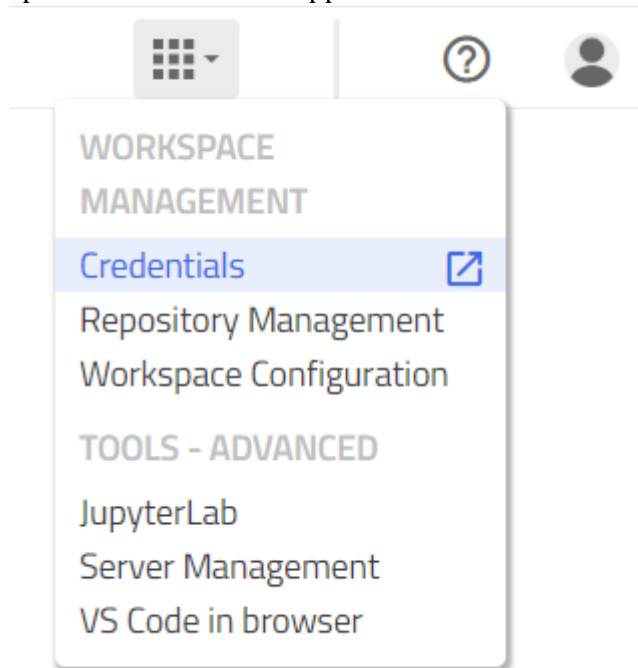
Setup

To get started you will need to go through the following steps:

1. Activate the credentials manager (Add a password for stored credentials)
2. Add credentials for a service (Store credentials for GitHub)
3. Check out a working copy
4. Create a workspace, using that working copy (including dependencies)

Go to the Modelon Impact landing page, and open the Apps menu in the top right.

1. Open the “Credentials” app:



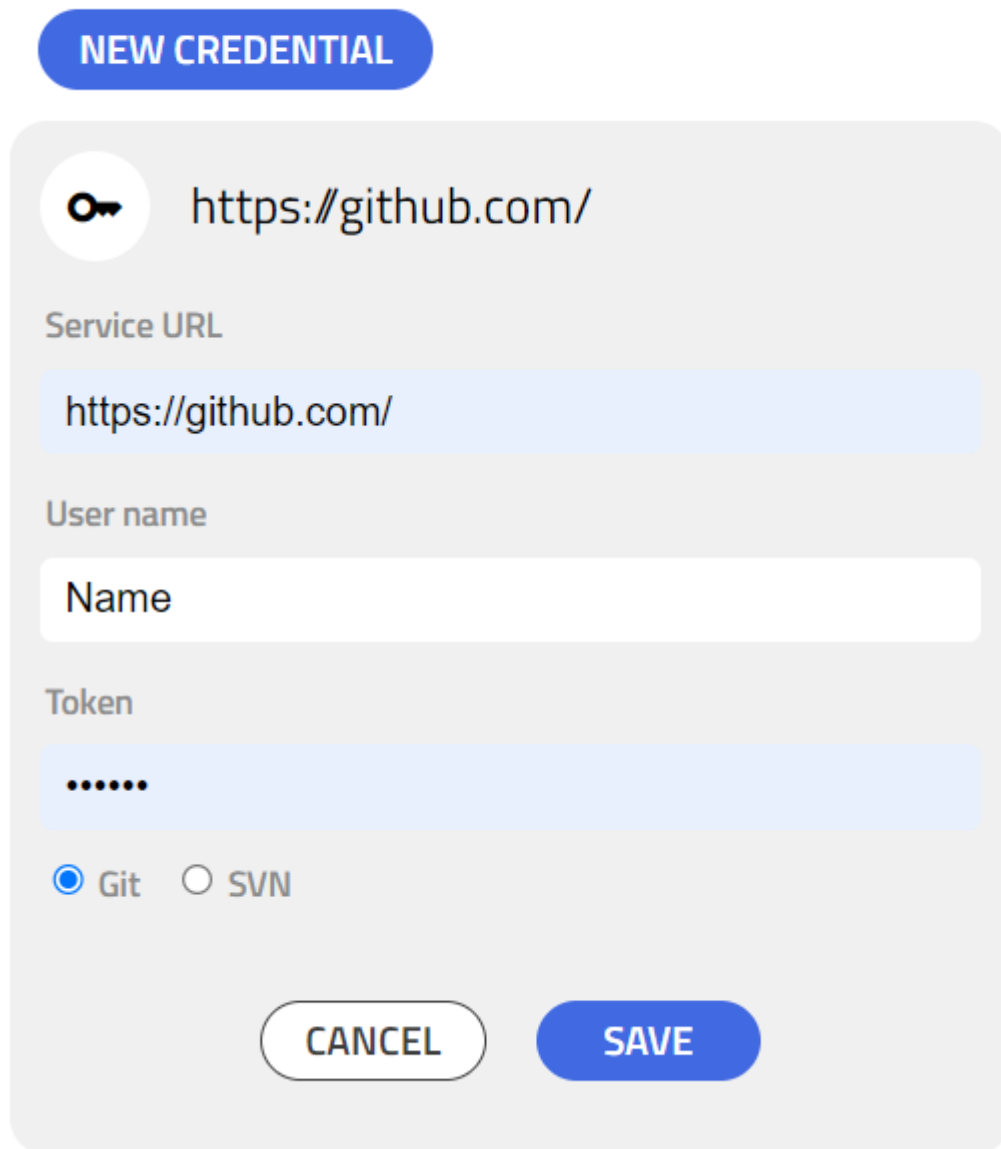
2. If this is the first time you activate the credentials manager, you need to set a password:
Note: this is a completely separate pwd from the Modelon Impact or the pwd used for any version control service, its just a pwd to protect the credentials used for the version control services.



You have not set up a credentials yet. Click [here](#) to set a password to start creating one

3. Click the link, and add a new pwd (remember it!), press enter.
Now we can add the credentials for the actual GitHub service.

4. Click “New Credential”:



NEW CREDENTIAL

https://github.com/

Service URL

https://github.com/

User name

Name

Token

.....

☒ Git ☐ SVN

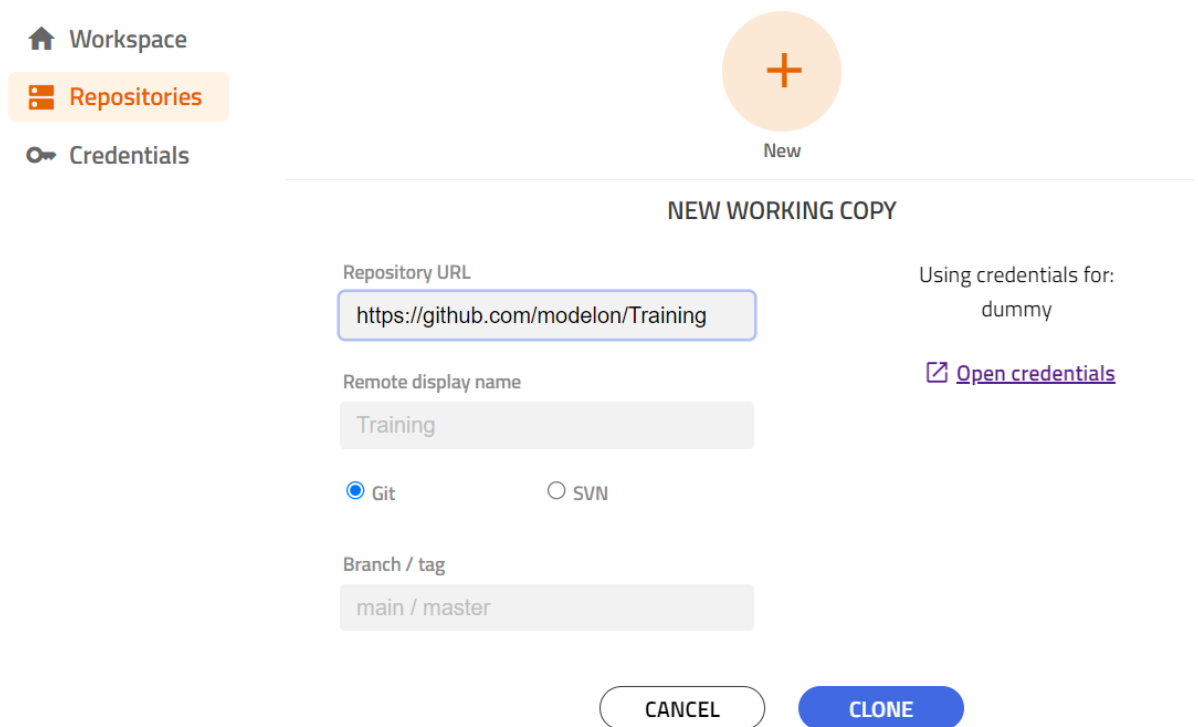
CANCEL SAVE

Note: For this exercise its not needed to add valid credentials, you will only create a copy of the repository.

If you want to commit/push content to the repository, you will need a GitHub account, add your credentials, and an API token (generated from your user management in GitHub).

Now its time to clone a local copy of the repository.

5. Open the “Repository” tab, and click “New”:



Workspace

Repositories

Credentials

New

NEW WORKING COPY

Repository URL

`https://github.com/modelon/Training`

Remote display name

Training

☒ Git ☐ SVN

Branch / tag

main / master

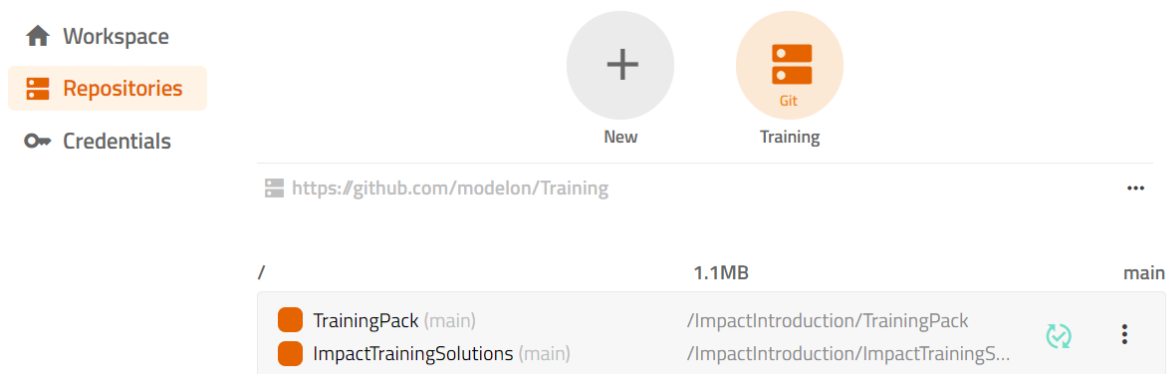
Using credentials for: dummy

[Open credentials](#)

CANCEL CLONE

6. Once filled in, click “Clone”

A local copy of all the content has now been created.



You can see on the green check box, that the local copy is consistent with the remote server.

Next step is to set up a workspace, that uses the Training content.

7. Click the Workspace tab, to create a new workspace:

Workspace

Repositories

Credentials

Workspace name

EDITABLE LIBRARIES

ImpactTrainingSolutions (main) ×

TrainingPack (main) ×

main

Drop libraries here to include them as editables

DEPENDENCIES

Electrification 1.8 ×

Hydraulics 4.17 ×

Modelon 4.1 ×

Drop libraries here to include them as dependencies

CREATE WORKSPACE

AVAILABLE LIBRARIES

AirConditioning 1.24

AircraftDynamics 1.7

EngineDynamics 2.9

EnvironmentalControl 3.13

FuelCell 1.15

FuelSystem 5.2

HeatExchanger 2.9

HydroPower 2.15

IndustryExamples 1.1

JetPropulsion 2.4

LiquidCooling 2.9

Pneumatics 2.13

ThermalPower 1.24

ThermoFluidPro 1.24

VaporCycle 2.9

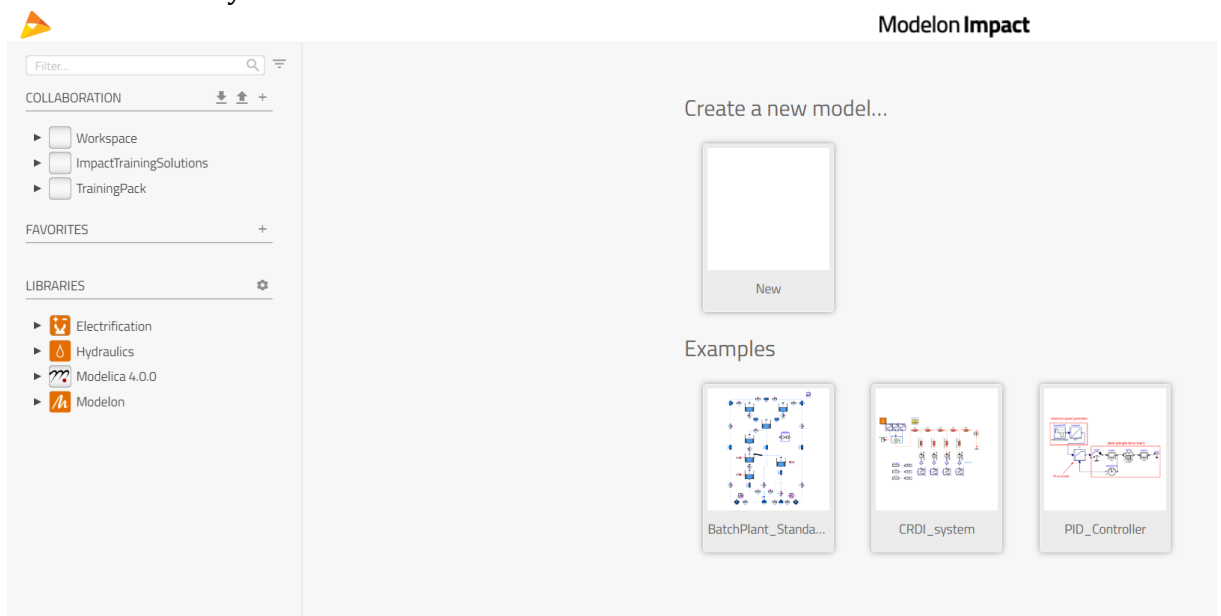
VehicleDynamics 4.1

[Clone/checkout new library](#)

- Click “Create Workspace” then open the workspace in Modelon Impact with the link at the bottom of the screen.

Open in [Modelon Impact](#) or [VS Code](#)

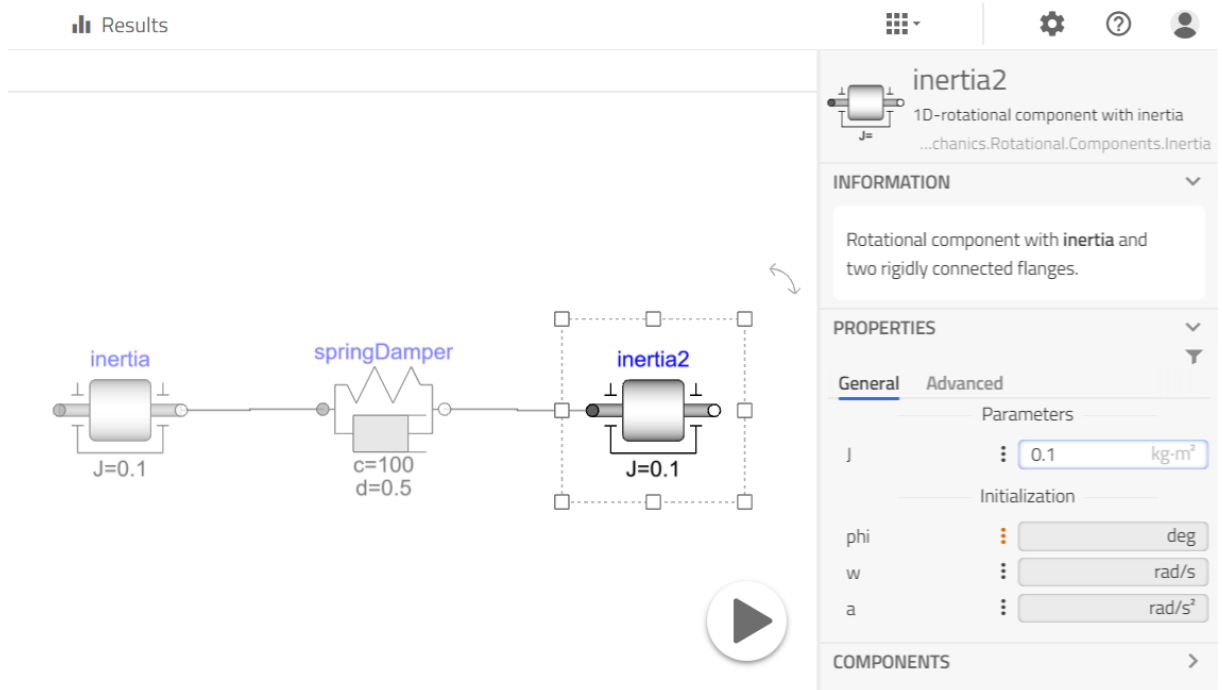
- You are now ready to start!



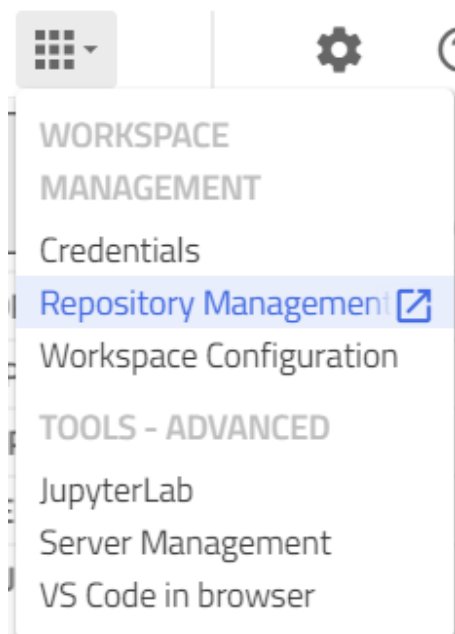
Modifying code

In the following section, we will start to author models, and see how the changes are tracked using version control.

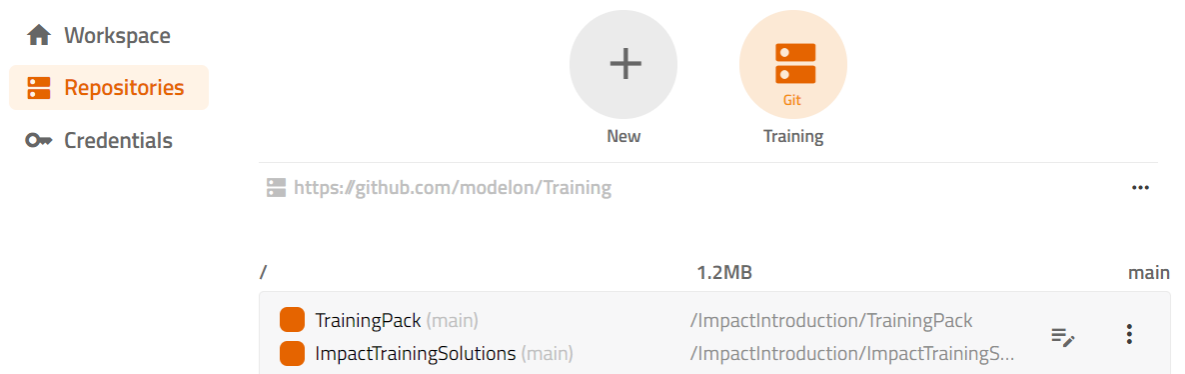
1. Open the model “ImpactTrainingSolutions.Day1.W1.MyFirstModel”.
2. Double click the component “inertia2”, and look at the properties:



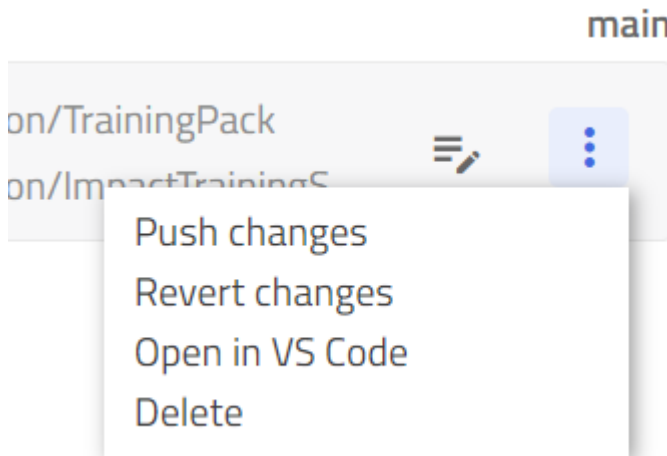
3. Here we can see that the chosen value for the inertia property is $J=0.1$.
Simulate the model, and plot the variable inertia2.phi.
4. Change the value to $J=0.2$, and simulate the model.
Plot the variable inertia2.phi, and compare the behavior change.
5. Now, let's see how the change is tracked in the version control system
 - a. First by opening the “Repository Management” App,
 - b. Then by opening the “VS Code in browser” App
6. Open the “Repository Management” App in the top right corner:



7. Here we can see that the status icon has been updated to indicate local changes:

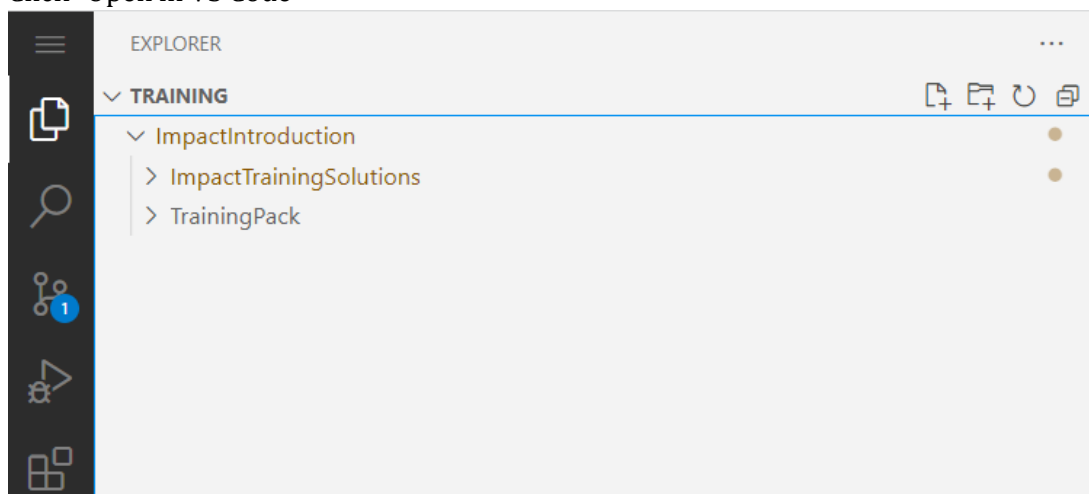


8. If we want to, we could push the changes directly to the repository:

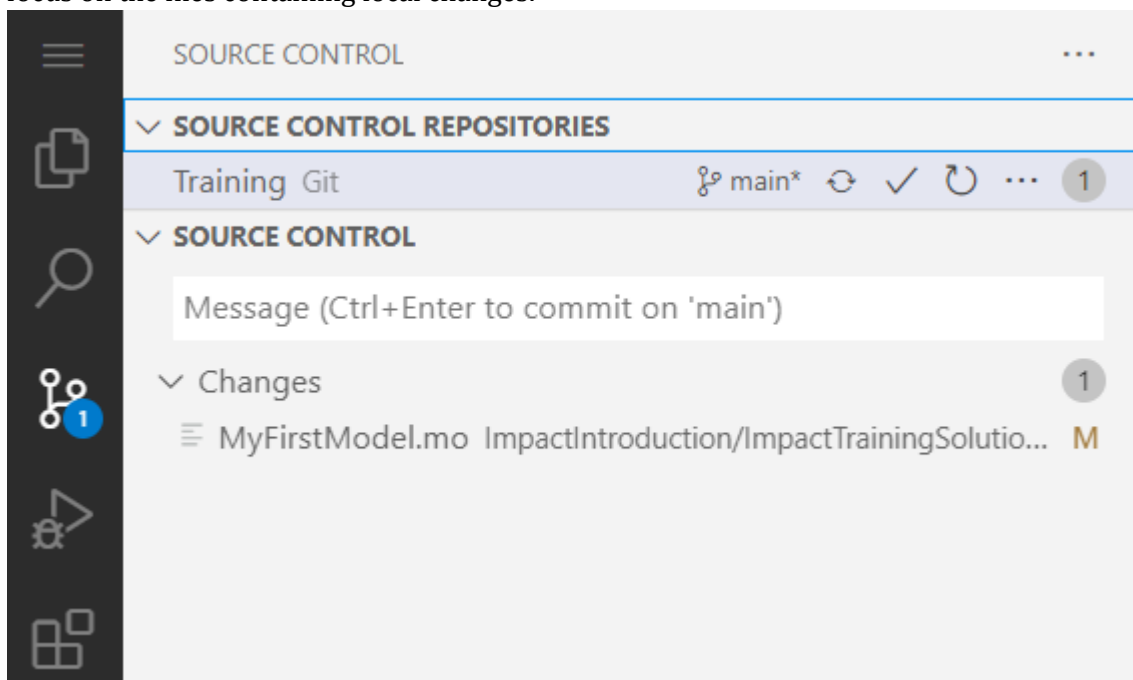


9. If we want a more detailed picture of the changes, we can use VS code. VS code also include a much more detailed interface to manage the full advanced workflows of version management.

10. Click “Open in VS Code”



11. VS Code is a very advanced tool with a vast array of capabilities, in particular we will look at the following:
 - a. File browser
 - b. File Editor
 - c. Version change tracing
12. As you can see, the “Explorer” is showing the cloned repository content used in the workspace.
13. Clicking the third Icon in the left toolbar, “Source Control”, that indicates 1 change, will focus on the files containing local changes.



14. Clicking the modified file allow you to have a look at the details of what changed:

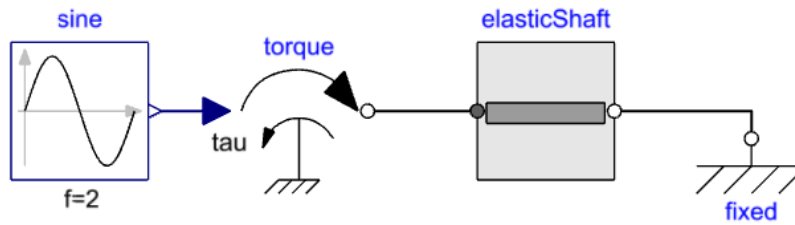


15. If you are fine with the changes, and you want to update the server, you need to go through the following 3 steps:
 - a. Stage (select what changes go together)
 - b. Commit (create a package to send over to the server, including comments on why you made the changes)
 - c. Push (Actually push the committed changes onto the server, in order to update the source files)
16. In the following workshop you don't have write access to the server, and you can safely revert the changes to get back to the original cloned copy version.
- 17.

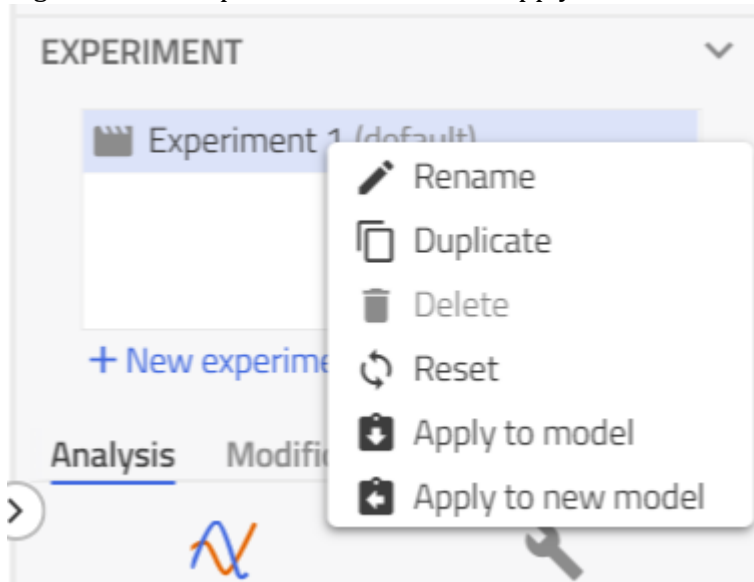
Advanced:

Open the model

"ImpactTrainingSolutions.Day1.W3.ComponentDesign.Experiments.TestShaft".



- Change a parameter in the Experiment 1.
- Run the experiment.
- Right click the experiment and choose "Apply to model".



- Open in VS code and examine the difference of where these two changes are stored in the workspace.
- Stage/commit and push (if applicable).

This concludes the workshop. Well done!