Introduction to Modelica and Thermo-fluid Modeling with Applications from the Buildings Library

Presented at the American Modelica Conference 2024

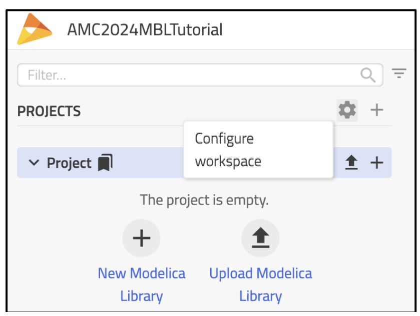
10/14/2024

This tutorial can utilize the Modelon Impact tool or OpenModelica as a Modelica development environment and simulation tool. To get started, use Initial A if using Impact or Initial B if using OpenModelica. The rest of the tutorial will use screenshots from Impact, however, OpenModelica can be used with the same components models and parameter settings.

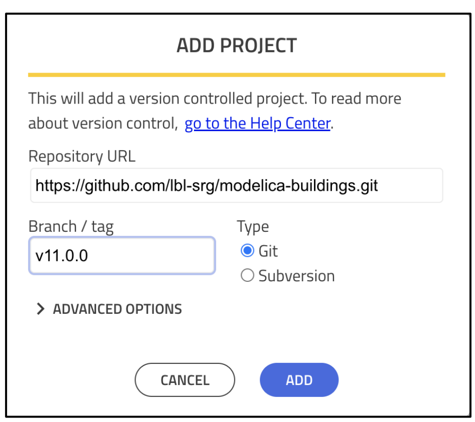
Initial A: Load Buildings Library into an Impact Project

Initially, create an Impact Workspace. Here, let's call it Tutorial. Before beginning the tutorial, we must first load the Modelica Buildings Library into that Workspace.

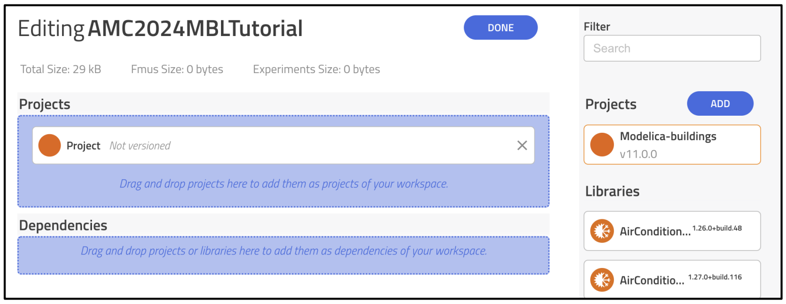
To do that, in Impact, expand the side menu to the left, click on the gear icon "Configure Workspace", and this will open a new browser tab.

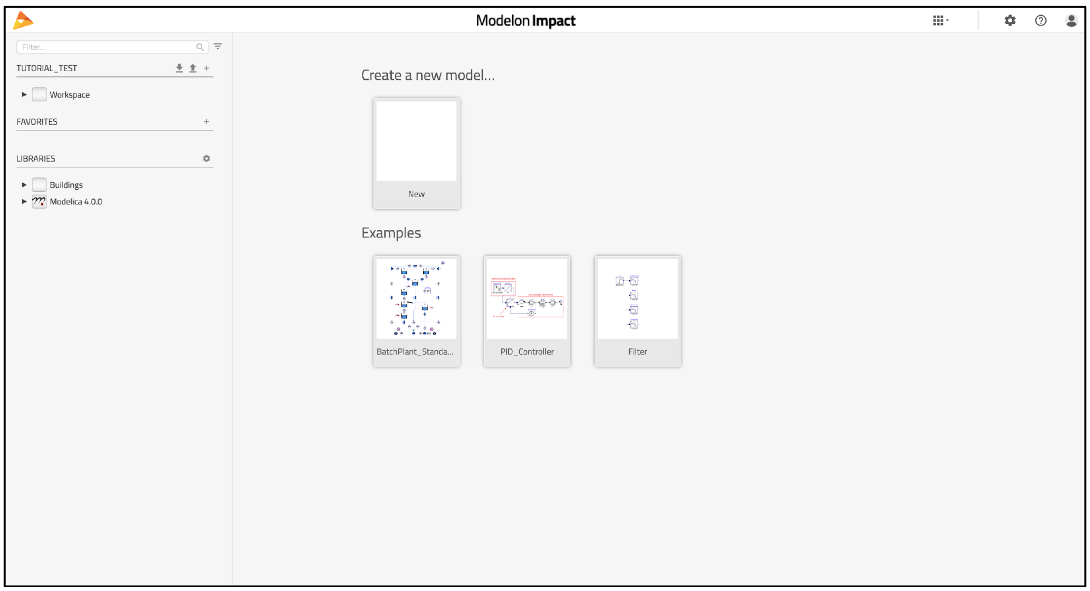


Go to Projects. Then, click "ADD PROJECT" and add the Repository URL for the Buildings Library (https://github.com/lbl-srg/modelica-buildings.git), and add a Branch/tag name of v11.0.0. This will use version 11.0.0 of the library. Then click ADD.

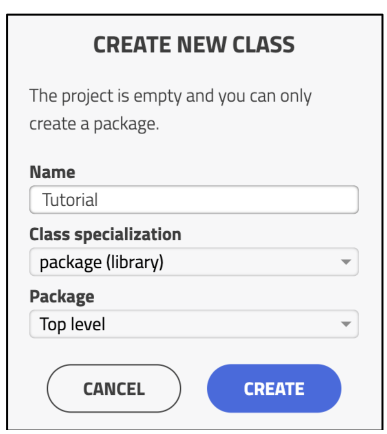


Then, go to Workspace and click Edit. Drag and drop Modelica-buildings from the Projects list into the Dependencies area of your workspace. Then, click DONE and reload your Workspace. You should see Buildings available in the Libraries list on the left.



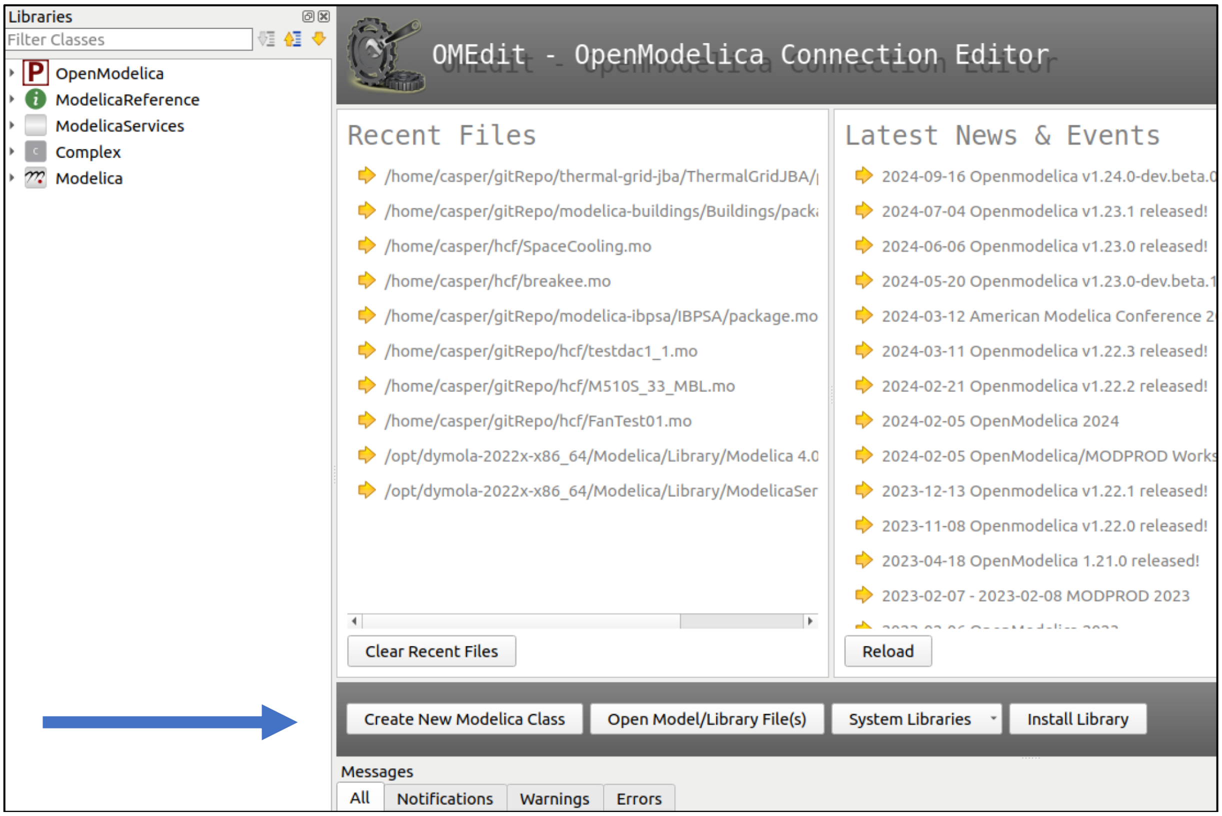


Finally, create a new Modelica library and call it "Tutorial". Now we have a new empty library to create models in.

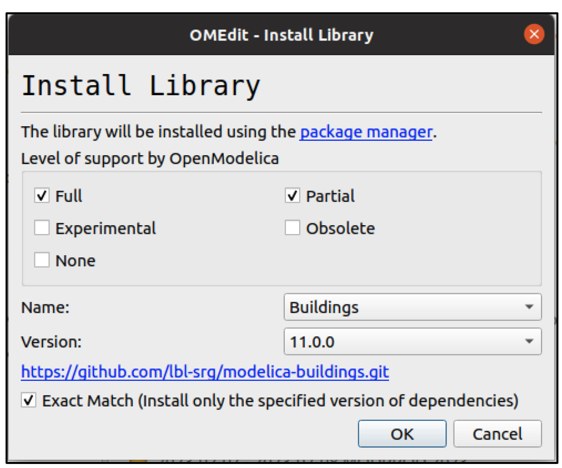


Initial B: Load Buildings Library into OMEdit for OpenModelica
This document was created at OpenModelica version 1.25.0~dev-24-g149ccc2.

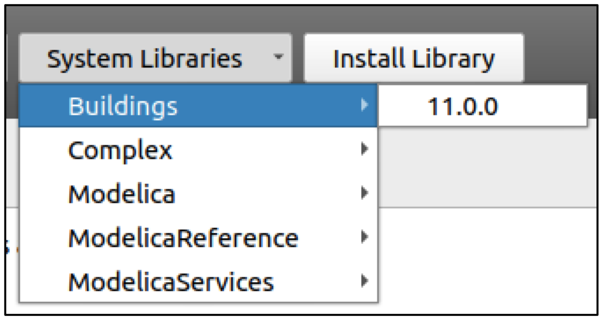
In OMEdit, locate the row of buttons close to the bottom of the window, above "Messages".



Click "Install Library". In the prompt, select "Building" in the dropdown list next to "Name" and make sure the version number is 11.0.0. Click "OK" to install Buildings library.
Alternatively, you can also go to Files > Manage Libraries > Install Library.



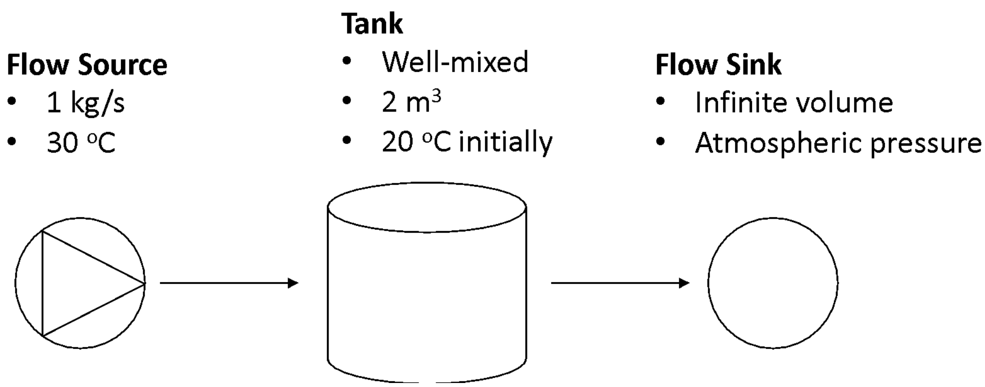
Finally, click “System Libraries” to load Buildings library. This button can also be found in Files > System Libraries.



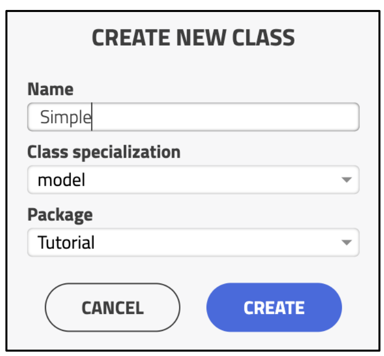
Now that the Building library is loaded, you can create models in OMEdit using its components.

Part I: Simple Thermofluid System

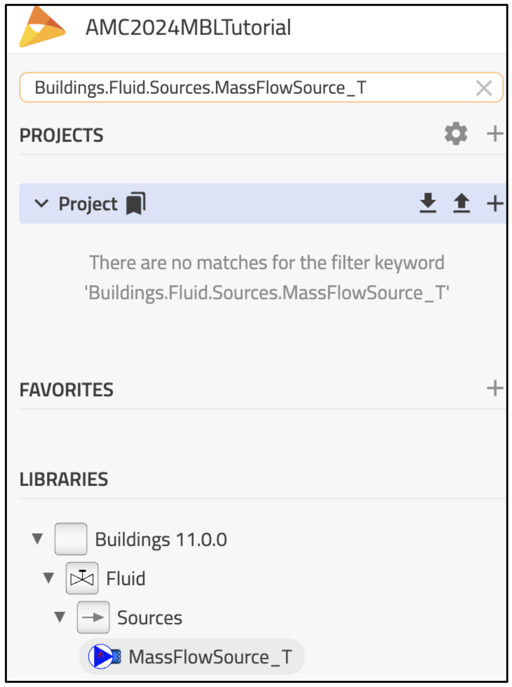
Let's start by implementing a simple thermofluid system model consisting of a water flow source of 1kg/s water at 30 ^{o}C, a well-mixed tank with no heat loss, a volume of 2 m3, and an initial temperature of 20 ^{o}C, and an infinite-volume flow sink that is at atmospheric pressure. A schematic of such a system is presented in the Figure below. It is maybe useful to think of the flow source as a pump with an infinite supply of water.



To implement the system model, first create a new model called "Simple." Note that you must have this class under a package (not "Top level") to be able to specify it as a model.



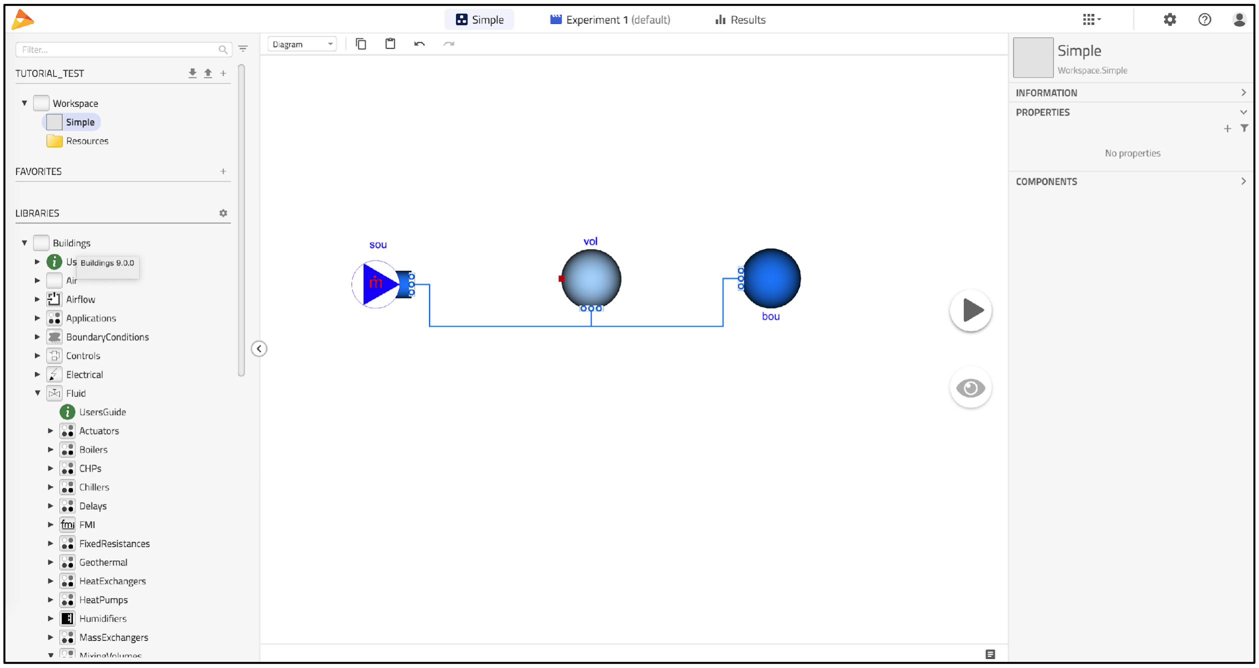
Then, drag and drop the component models defined in the table below. You can find the components more quickly by using the search bar. Then, parameterize them as described in the table. Finally, connect the components as you see in the diagram.

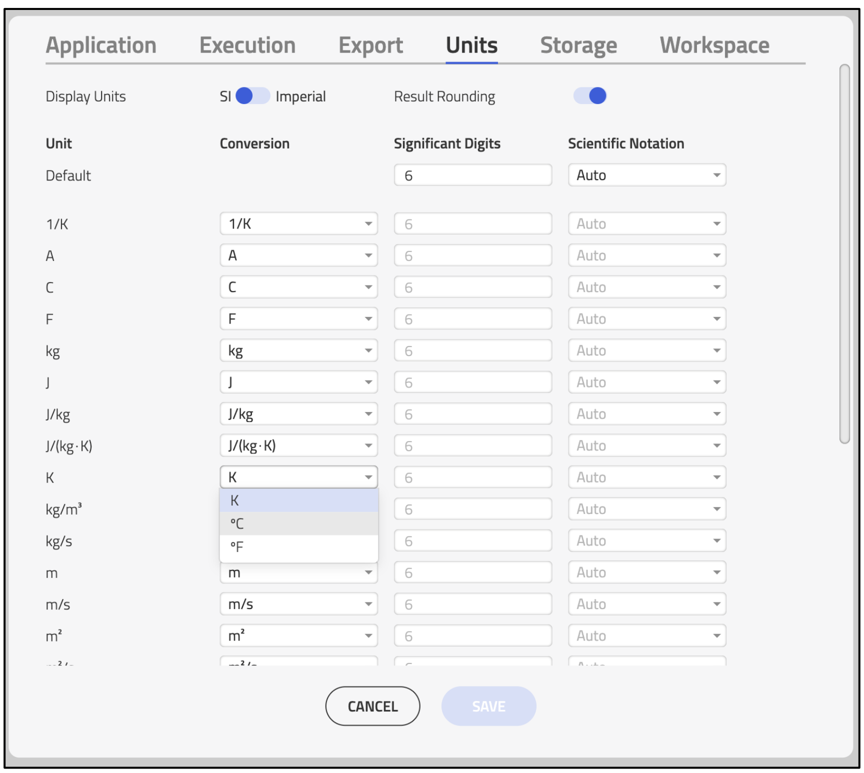


We use three component models for an ideal flow source, a mixing volume, and a flow boundary condition. Then, we'll use what is known as a media model (or medium), which is a collection of algebraic equations for thermodynamic variables used to specify to each of the first three components that water is flowing through them (as opposed to air or some other fluid).

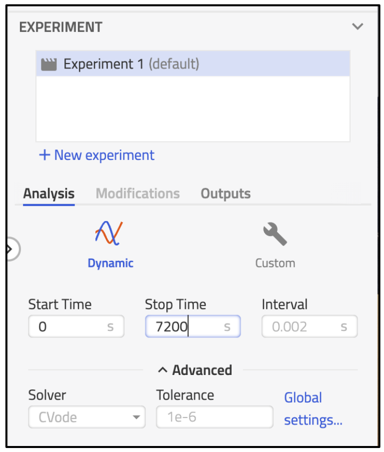
Temperature parameters are defined below in Kelvin, but you can change the unit settings in Impact (Settings Gear > Units) to specify temperatures in °C. This will also make result plots of temperature variables in °C later on.

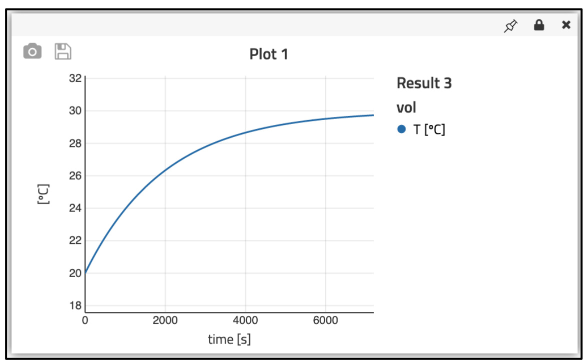
Component Model	Name	Description	Parameters
Buildings.Fluid.Sources.MassFlowSource_T	sou	Ideal flow source	General • Medium=Water • m_flow=1 • T=273.15+30
Buildings.Fluid.Sources.Boundary_pT	sin	Infinite flow sink	• Medium=Water
Buildings.Fluid.MixingVolumes.MixingVolume	vol	Mixing volume	• Medium=Water • V=2 • m_flow_nominal=1


Simple
Experiment 1 (default)
Results
TUTORIAL_TEST
Workspace
Simple
Resources
FAVORITES
LIBRARIES
Buildings
Airflow
Applications
BoundaryConditions
Controls
Electrical
Fluid
UsersGuide
Actuators
Boilers
CHPs
Chillers
Delays
FMI
FixedResistances
Geothermal
HeatExchangers
HeatPumps
Humidifiers
MassExchangers
Diagram
sou
vol
bou
Simple
Workspace.Simple
INFORMATION
PROPERTIES
No properties
COMPONENTS


Application
Execution
Export
Units
Storage
Workspace
Display Units
SI
Imperial
Result Rounding
Unit
Conversion
Significant Digits
Scientific Notation
Default
6
Auto
1/K
A
C
F
kg
J
J/kg
J/(kg·K)
K
kg/m³
kg/s
m
m/s
m²
K
°C
°F
CANCEL
SAVE

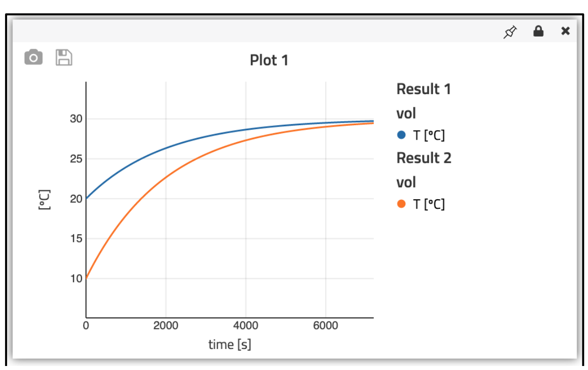
Now let's simulate the model for 2 hours and see how the temperature of the water in the tank changes over time. Note that by default, the water in the tank starts at 20 °C. Set up the experiment so the start time is 0 s and final time is 7200 s using the Experiment tab. Then, execute the simulation using the play button and plot the results by clicking and dragging the variable "vol.T" onto the canvas.





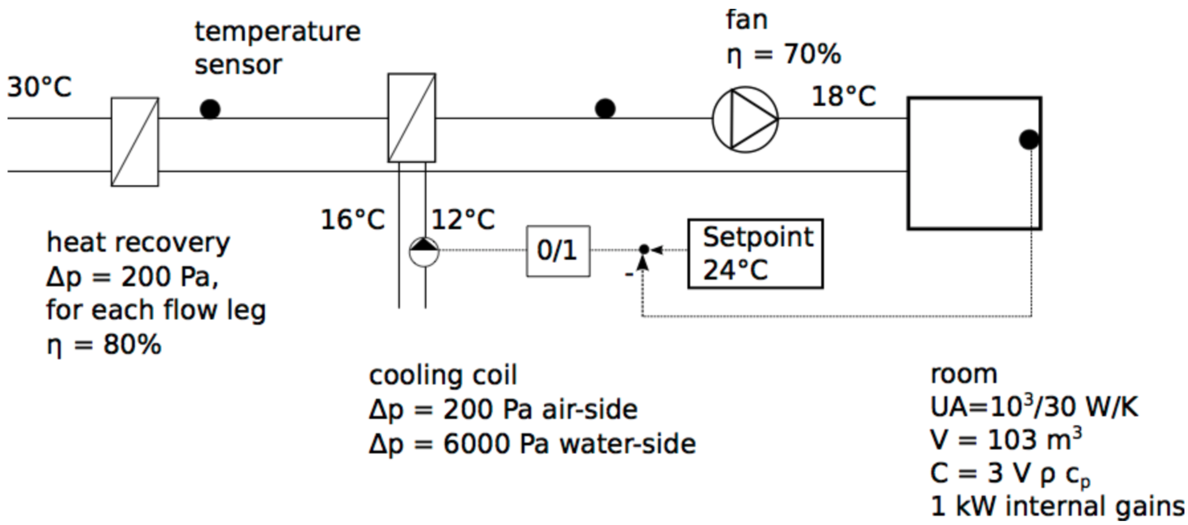
Advanced:

Change the initial temperature of the water in the tank to 10 °C by changing the parameter "T_start" to 10 °C in the Initialization tab in the component "vol." Re-simulate and see the new results on the same plot as the old results to compare (click and drag the variable "vol.T" from the new result to the existing plot).



Part II: Space Cooling Tutorial

This tutorial is adopted from the Modelica Buildings Library. We will implement a space cooling system with supply fan, cooling coil, fresh air supply with heat recovery, and space temperature control. We will also implement a schedule-based temperature control set point that emulates a demand response event. The system diagram is shown in below.

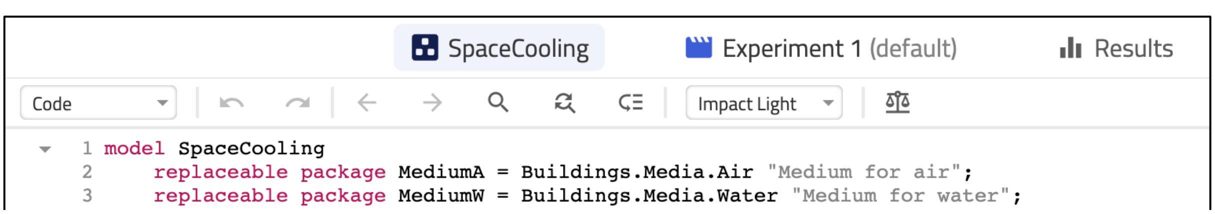


1. Room Model

We will first implement a simple room model, represented by a volume of air with an enlarged heat capacity to account for internal thermal mass in furniture and building constructions, a prescribed heat source for the internal convective heat gain, and a heat conductor for steady-state heat conduction to the outside.

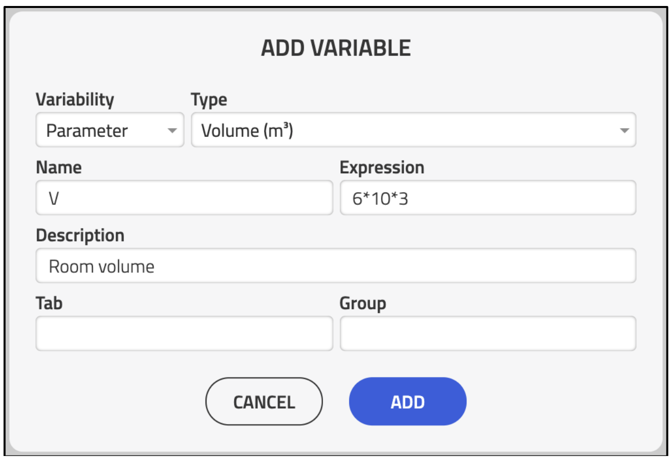
Create a new model called “SpaceCooling.” Declare a moist air media model and a water model at the top level to be able to propagate to lower-level component models later in this exercise. Declare the media model by switching to Code view and then writing the appropriate declaration as follows:

```
replaceable package MediumA = Buildings.Media.Air "Medium for air";
replaceable package MediumW = Buildings.Media.Water "Medium for water";
```



Next, we will declare system-level parameters for the room volume, nominal air mass flow rate, and internal heat gains of the room. These system-level parameters will be propagated down to lower-level models. Declare the system-level parameters using the "Properties > Add variable (Plus icon)" according to the table below. Notice parameters can be used in expressions of other parameters.

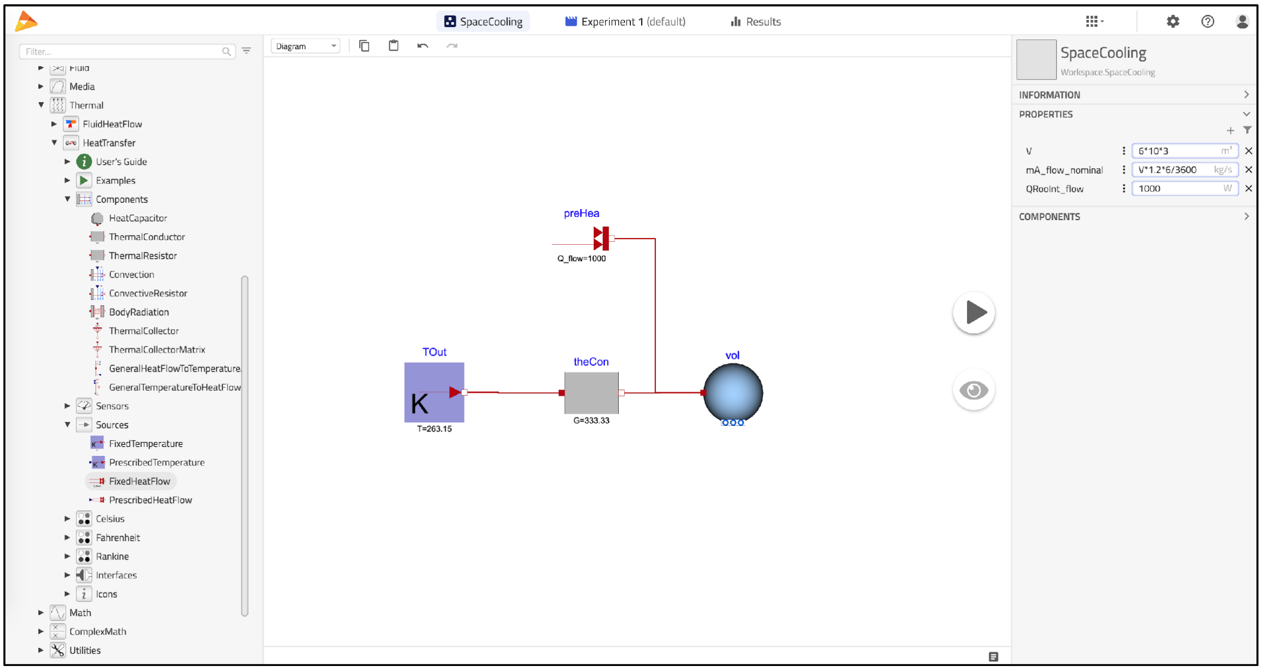
Variability	Type	Name	Expression	Description
Parameter	Volume	V	6*10*3	Room volume
Parameter	MassFlowRate	mA_flow_nominal	V*1.2*6/3600	Nominal mass flow rate
Parameter	HeatFlowRate	QRooInt_flow	1000	Internal heat gains of the room



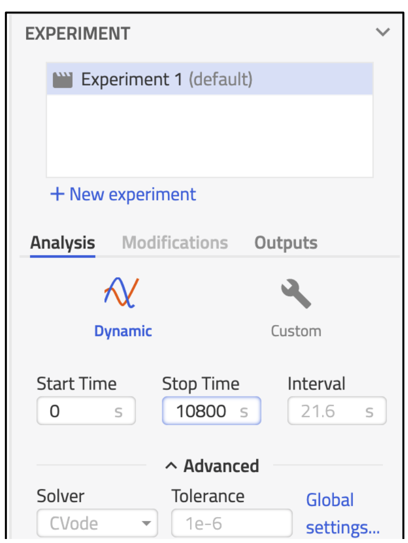
Now let's implement our room model. Drag and drop the component models defined in the table below. Then, parameterize them as described in the table. Finally, connect the components as you see in the diagram. Note for the Medium declaration you need to type MediumA.

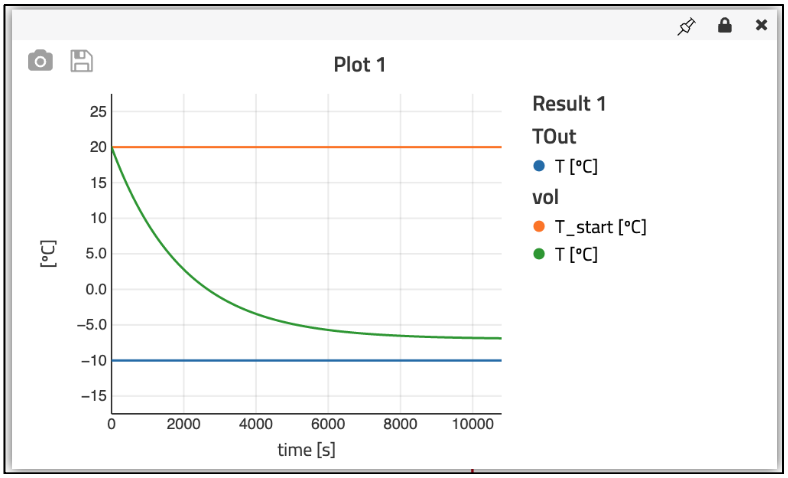
Component Model	Name	Description	Parameters
Buildings.Fluid.MixingVolumes.MixingVolume	vol	Room air volume	<u>General</u> • Medium = MediumA • V=V • m_flow_nominal=mA_flow_nominal <u>Dynamics</u> • energyDynamics=Modelica.Fluid.Types.Dynamics.FixedInitial • mSenFac=3
Modelica.Thermal.HeatTransfer.Components.ThermalConductor	theCon	Thermal conductance with the ambient	<u>General</u> • G=10000/30

Modelica.Thermal.HeatTransfer.Sources.FixedTemperature	TOut	Outside temperature	• T=273.15-10
Modelica.Thermal.HeatTransfer.Sources.FixedHeatFlow	preHea	Prescribed internal heat gain flow rate	• Q_flow=QRooInt_flow



Finally, let's simulate the model for 3 hours, or 10800 seconds, and observe the response. First, edit the "Stop Time" in the "Experiment" tab. Then, simulate the model and view the results.





Plot 1
25
20
15
10
5.0
0.0
−5.0
−10
−15
[°C]
0
2000
4000
6000
8000
10000
time [s]
Result 1
TOut
T [°C]
vol
T_start [°C]
T [°C]

2. System Model

Now we're ready to implement the system model, operating under open-loop control. If you have not completed section 1. Room Model, start by copying the model "Buildings.Examples.Tutorial.SpaceCooling.System1." Then define additional top-level parameters as defined in the table below.

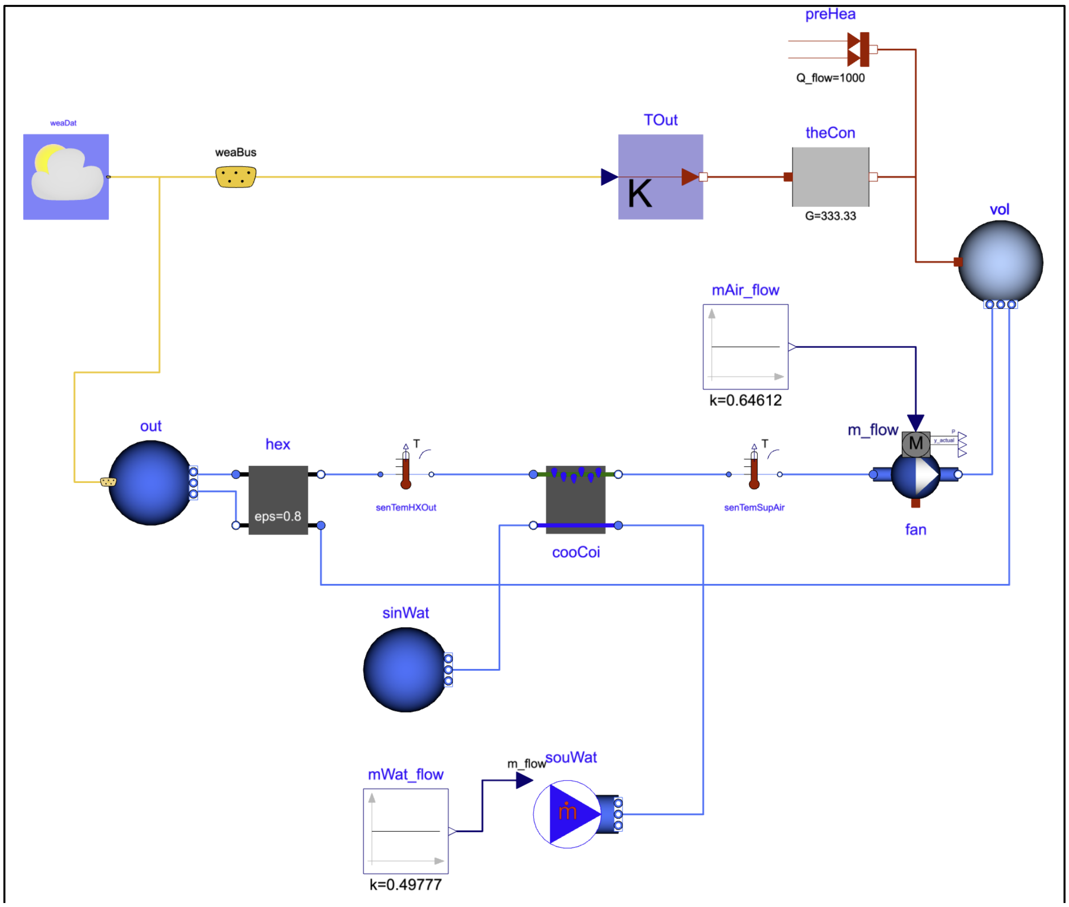
Variability	Type	Name	Expression	Description
Parameter	Real	eps	0.8	Heat recovery effectiveness
Parameter	Temperature	TRooSet	273.15+24	Nominal room air temperature
Parameter	Temperature	TASup_nominal	273.15+18	Nominal air temperature supplied to room
Parameter	Dimensionless Ratio	wASup_nominal	0.012	Nominal supply air humidity ratio [kg/kg]
Parameter	Temperature	TOut_nominal	273.15+30	Design outside air temperature
Parameter	Temperature	THeaRecLvg	TOut_nominal - eps*(TOut_nominal-TRooSet)	Nominal air temperature leaving the heat recovery
Parameter	Dimensionless Ratio	wHeaRecLvg	0.0135	Nominal air humidity ratio [kg/kg] leaving the heat recovery
Parameter	HeatFlowRate	QRooC_flow_nominal	-QRooInt_flow-10E3/30*(TOut_nominal-TRooSet)	Nominal cooling load of the room
Parameter	MassFlowRate	mA_flow_nominal * Note this is a change to an existing parameter	1.3*QRooC_flow_nominal/1006/(TASup_nominal-TRooSet)	Nominal air mass flow rate, increased by factor 1.3 to allow for recovery after temperature setback
Parameter	TemperatureDifference	dTFan	2	Estimated temperature raise across fan that needs to be made up by the cooling coil
Parameter	HeatFlowRate	QCoiC_flow_nominal	mA_flow_nominal*(TASup_nominal-THeaRecLvg-dTFan)*1006+mA_flow_nominal*(wASup_nominal - wHeaRecLvg)*2458.3e3	Cooling load of coil, taking into account economizer, and increased due to latent heat removal
Parameter	Temperature	TWSup_nominal	273.15+12	Water supply temperature
Parameter	Temperature	TWRet_nominal	273.15+16	Water return temperature
Parameter	MassFlowRate	mW_flow_nominal	-QCoiC_flow_nominal / (TWRet_nominal-TWSup_nominal)/4200	Nominal water mass flow rate

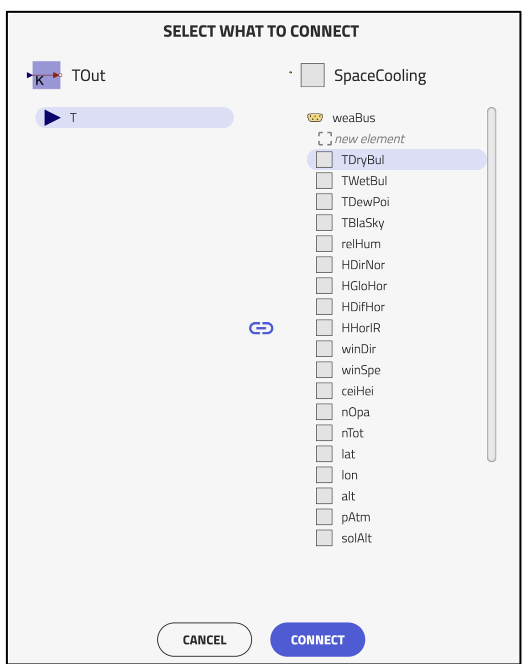
Now let's implement our system model. First, replace the component "TOut" with a "Modelica.Thermal.HeatTransfer.Sources.PrescribedTemperature." Then, drag and drop the additional component models defined in the table below. Then, parameterize them as described in the table. Finally, connect the components as you see in the diagram.

Pay special attention to the arrangement of the cooling coil model such that port a2 is towards the top left of the component. Also note that connecting the weather bus from the component "weaDat" to "TOut", type the variable from the bus wanting to be connected, "TDryBul," as shown in the screen shot below.

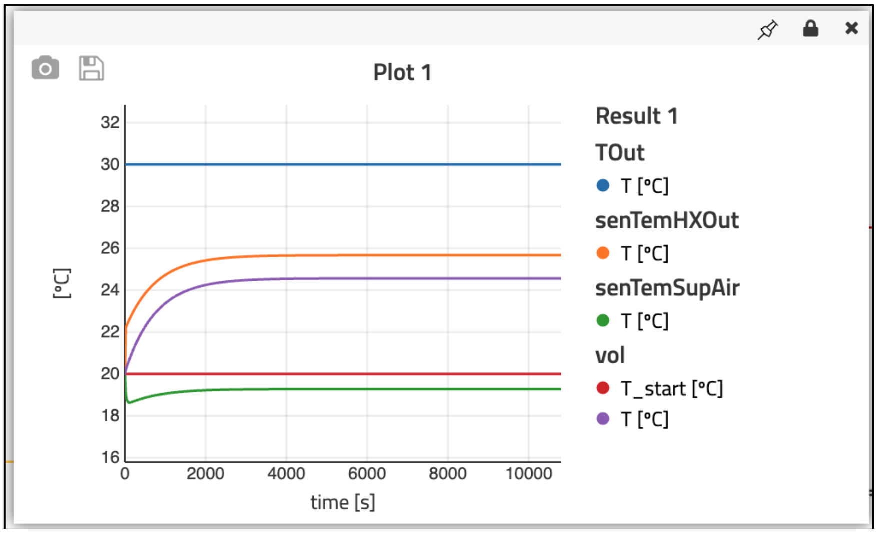
Component Model	**Name**	**Description**	**Parameters**
Buildings.Fluid.Movers.FlowControlled_m_flow	fan	Supply air fan	• Medium=MediumA • m_flow_nominal=mA_flow_nominal Dynamics • energyDynamics= Modelica.Fluid.Types.Dynamics.SteadyState
Buildings.Fluid.HeatExchangers.ConstantEffectiveness	hex	Heat recovery	• Medium1=MediumA • Medium2=MediumA • m1_flow_nominal=mA_flow_nominal • m2_flow_nominal=mA_flow_nominal • dp1_nominal=200 • dp2_nominal=200
Buildings.Fluid.HeatExchangers.WetCoilEffectivenessNTU	cooCoi	Cooling coil	• Medium1=MediumW • Medium2=MediumA • m1_flow_nominal=mW_flow_nominal • m2_flow_nominal=mA_flow_nominal • dp1_nominal=6000 • dp2_nominal=200 • use_Q_flow_nominal=true *(tick the box)* • Q_flow_nominal= QCoiC_flow_nominal • T_a1_nominal=TWSup_nominal • T_a2_nominal=THeaRecLvg • w_a2_nominal= wHeaRecLvg Advanced • show_T=true *(tick the box)* Dynamics • energyDynamics=Modelica.Fluid.Types.Dynamics.FixedInitial
Buildings.Fluid.Sources.Outside	out	Ambient air source	• Medium=MediumA
Buildings.Fluid.Sources.MassFlowSource_T	souWat	Source for water flow rate	• Medium=MediumW • use_m_flow_in=true *(tick the box)* • T=TWSup_nominal
Buildings.Fluid.Sources.Boundary_pT	sinWat	Sink for water circuit	• Medium=MediumW

Buildings.BoundaryConditions.WeatherData.ReaderTMY3	weaDat	Weather data reader	● filNam=Modelica.Utilities.Files.loadResource("modelica://Buildings/Resources/weatherdata/USA_IL_Chicago-OHare.Intl.AP.725300_TMY3.mos")) *(May need to edit this in the text view.)* Data source ● pAtmSou=Buildings.BoundaryConditions.Types.DataSource.Parameter ● TDryBulSou=Buildings.BoundaryConditions.Types.DataSource.Parameter ● TDryBul=TOut_nominal
Buildings.BoundaryConditions.WeatherData.Bus	weaBus	Weather bus connector	
Buildings.Controls.OBC.CDL.Reals.Sources.Constant	mAir_flow	Fan air flow rate	k=mA_flow_nominal
Buildings.Controls.OBC.CDL.Reals.Sources.Constant	mWat_flow	Water flow rate	k=mW_flow_nominal
Buildings.Fluid.Sensors.TemperatureTwoPort	senTemHXOut	Temperature sensor for heat recovery outlet on supply side	● Medium=MediumA ● m_flow_nominal=mA_flow_nominal
Buildings.Fluid.Sensors.TemperatureTwoPort	senTemSupAir	Temperature sensor for supply air	● Medium=MediumA ● m_flow_nominal=mA_flow_nominal


preHea
Q_flow=1000
TOut
theCon
G=333.33
weaDat
weaBus
vol
mAir_flow
k=0.64612
m_flow
out
hex
eps=0.8
senTemHXOut
cooCoi
senTemSupAir
fan
sinWat
mWat_flow
k=0.49777
m_flow
souWat


SELECT WHAT TO CONNECT
TOut
SpaceCooling
T
weaBus
new element
TDryBul
TWetBul
TDewPoi
TBlaSky
relHum
HDirNor
HGloHor
HDifHor
HHorIR
winDir
winSpe
ceiHei
nOpa
nTot
lat
lon
alt
pAtm
solAlt
CANCEL
CONNECT

Finally, let's simulate the model for the same 3 hours as previously. Double check the Experiment is set up so "Start Time" is 0 s and "Stop Time" is 10800 s. Then, simulate the model and view the results.

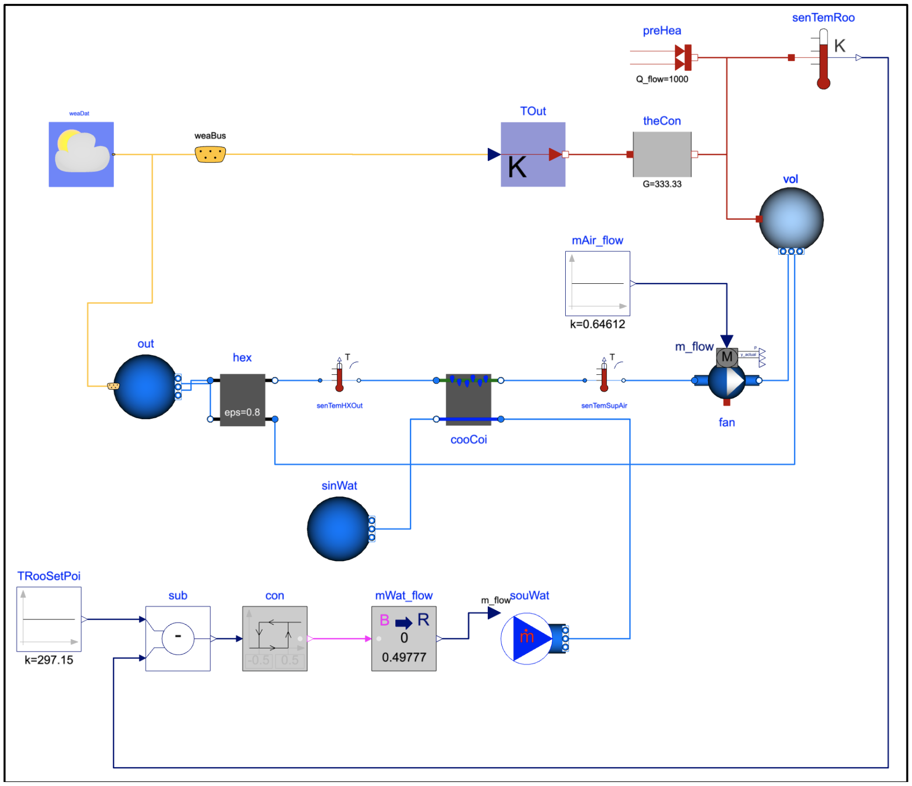


3. Closed Loop Control

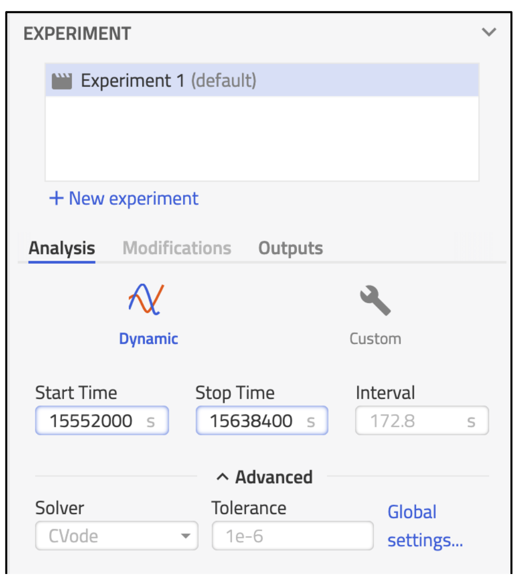
Now let's simulate the system in more realistic ambient conditions and add feedback control in the form of an on/off controller based on room temperature measurement controlling the flow of chilled water. If you did not complete 2. System Model, then start by copying "Buildings.Examples.Tutorial.SpaceCooling.System2."

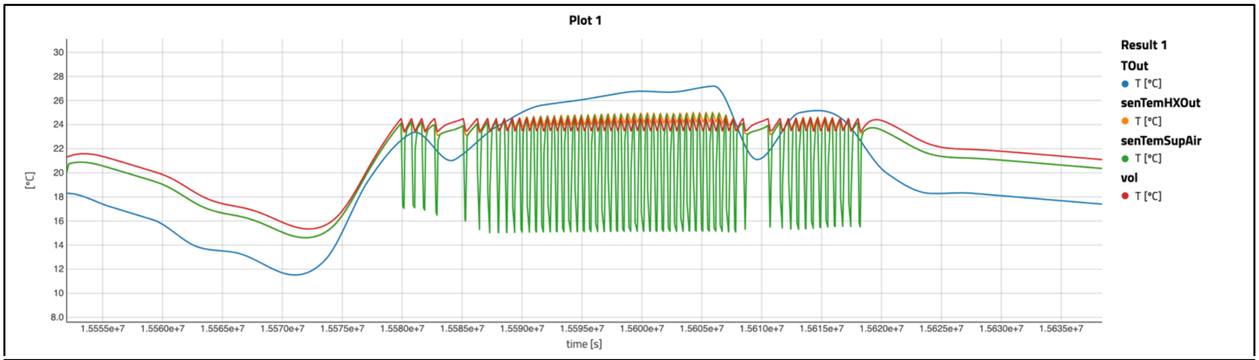
First, let's change the source of the outside dry bulb temperature to the weather file instead of a constant value. Do this by selecting "Use data from file" for the parameter "weaDat.TDryBulSou." Then, add control by dragging and dropping the additional component models defined in the table below. Then, parameterize them as described in the table. Finally, connect the components as you see in the diagram.

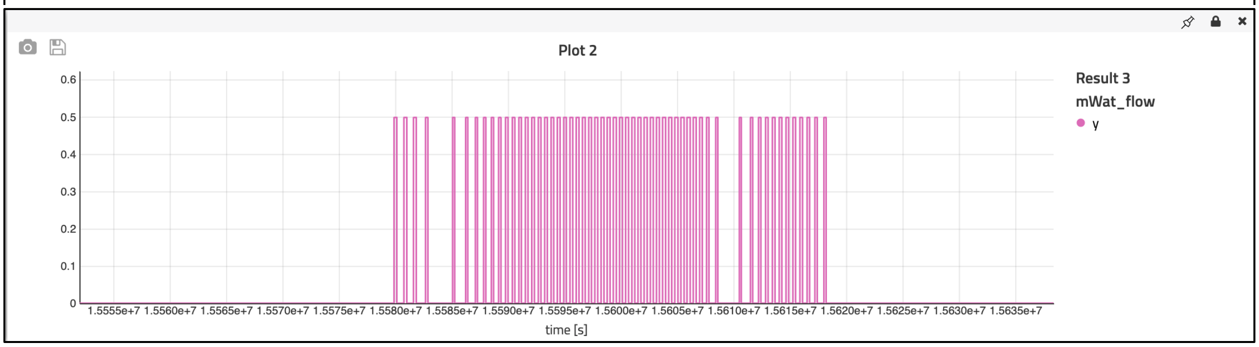
Component Model	**Name**	**Description**	**Parameters**
Buildings.Controls.OBC.CDL.Reals.Hysteresis	con	Controller for coil water flow rate	• bandwidth=1
Buildings.Controls.OBC.CDL.Reals.Subtract	sub	Subtraction	•
Buildings.Controls.OBC.CDL.Reals.Sources.Constant	TRooSetPoi	Room temperature set point	• k=TRooSet
Modelica.Thermal.HeatTransfer.Sensors.TemperatureSensor	senTemRoo	Room temperature sensor	
Buildings.Controls.OBC.CDL.Conversions.BooleanToReal *Replacing Buildings.Controls.OBC.CDL.Continuous.Sources.Constant	mWat_flow	Conversion from boolean to real for water flow rate	• realTrue=0 • realFalse=mW_flow_nominal



Finally, let's simulate the model for one day during summer from hour 4320 to 4344, and observe the response. First, edit the "Start Time" and "Stop Time" in the "Experiment" tab. Then, simulate the model and view the results.




Plot 1
Result 1
TOut
T [°C]
senTemHXOut
T [°C]
senTemSupAir
T [°C]
vol
T [°C]
[°C]
time [s]

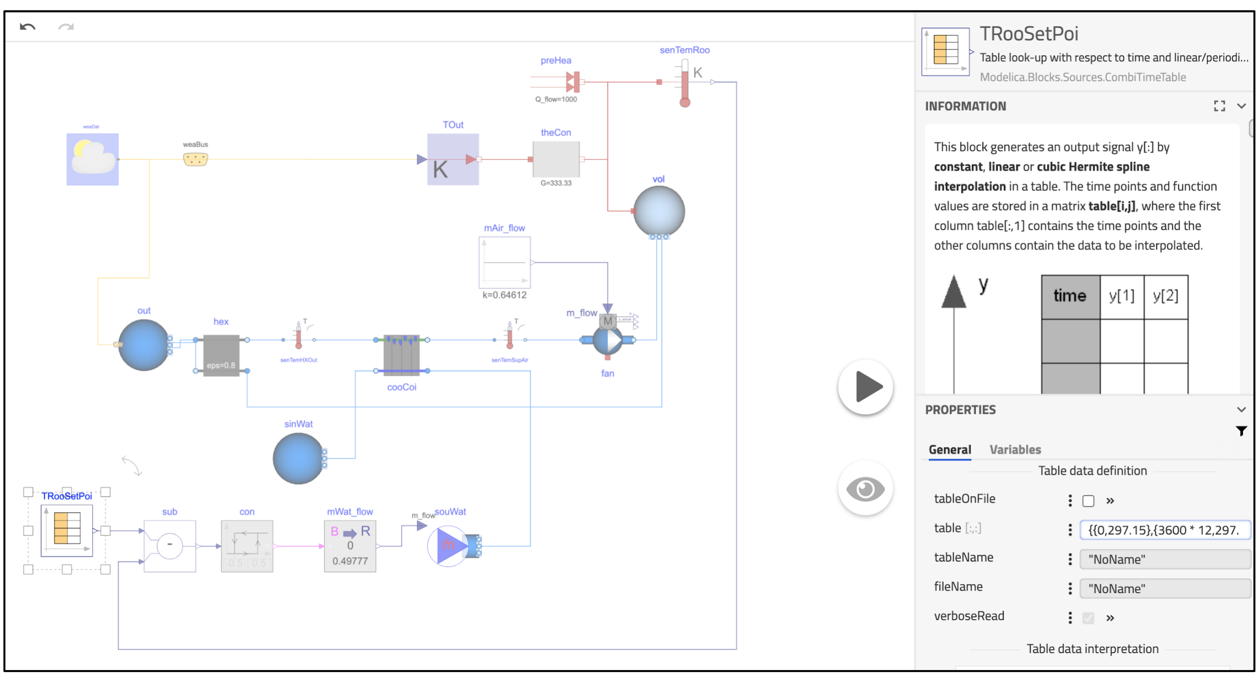

Plot 2
Result 3
mWat_flow
y
time [s]

4. Demand Response

Now, let's use our system to simulate a demand response event, where the room set point temperature is increased by 2 °C for three hours in the afternoon in order to reduce energy usage during that time. This may be valuable to an electric grid that is strained during this time.

Start by replacing the constant room temperature set point with a schedule-based set point. To do this, replace the existing "TRooSetPoi" with the component "Modelica.Blocks.Sources.CombiTimeTable." Then, set the parameters of this component as follows in the table below. Note that the "Table" parameter sets the schedule of a typical day in pairs of {time [sec], temperature [K]}. Then, the "smoothness" parameter ensures the values are implemented stepwise in time and constant between the defined points. Finally, the "extrapolation" parameter ensures the same schedule is repeated every day (24 hours).

Component Model	Name	Description	Parameters
Modelica.Blocks.Sources.CombiTimeTable *Replacing Buildings.Controls.OBC.CDL.Continuous.Sources.Constant	TRooSetPoi	Schedule-based room temperature set point	● table = {{0,297.15},{3600*12,297.15},{3600*15,297.15},{3600*24,297.15}} ● smoothness = ConstantSegments ● extrapolation = Periodic



Simulate the model to obtain a baseline result. Then, change the temperature set point at hour 12 to 299.15 (26 C)* and re-simulate. Plot the results and compare the room temperature and cooling water mass flow profiles. What do you notice?

* table = {{0,297.15},{3600*12,299.15},{3600*15,297.15},{3600*24,297.15}}