Model Commissioning Walkthrough

BuildingMOTIF Model Commissioning Workflow Tutorial

This web application is a tutorial tool for the BuildingMOTIF software development kit (SDK). The goal of this tutorial is to demonstrate some key parts of the BuildingMOTIF model commissioning workflow.

1. Define a naming convention for the building's point labels, or upload an existing CSV
2. Upload point schedules create templates which will be used to create a metadata model of the building
3. Upload a point list to create a model

Proceed by clicking on each of the links below, in order. Start with Load Point Schedules.

Setup Start Here

Define Naming Convention

Create Templates from Point Schedules

Setup (Advanced)

Configure Point Label Parser

Check Point Label Parse

Create Metadata Model

From Existing TTL file

From Existing Point Dump file

Model Validation

Create Model Manifest

Validate Most Recent Model (urn:buidling1)

Application Configuration

Generate FDD Rule Implementation Report

Database

Libraries

Models

Templates

If the UI is bugged, hit this button: Clear Demo State

Mappings

This page is for managing mappings from point abbreviations and descriptions to Brick classes. You can add new mappings manually by clicking "Add Row", or upload a CSV file of existing mappings. The "Suggest" buttons (the lightbulb icon) will use a machine learning model to guess the appropriate Brick classes for a given description. Click "Suggest for Blanks" to fill in all missing point and equipment classes. When you are finished, click "Save Mappings" to save your work.

Demo: We recommend uploading the `finished_mapping.csv` file provided with this release as an example.

Add Row | Upload CSV | Download CSV | Save Mappings | Suggest for Blanks

Upload the “finished_mapping.csv” file we provide

Abbreviation	Description	Brick Point Class	Brick Equip Class	Brick Location Class

Home | Mapping | Models | Libraries | FDD Rules

Mappings

This page is for managing mappings from point abbreviations and descriptions to Brick classes. You can add new mappings manually by clicking "Add Row", or upload a CSV file of existing mappings. The "Suggest" buttons (the lightbulb icon) will use a machine learning model to guess the appropriate Brick classes for a given description. Click "Suggest for Blanks" to fill in all missing point and equipment classes. When you are finished, click "Save Mappings" to save your work.

Demo: We recommend uploading the `finished_mapping.csv` file provided with this release as an example.

Add Row | Upload CSV | Download CSV | Save Mappings | Suggest for Blanks

Abbreviation	Description	Brick Point Class	Brick Equip Class	Brick Location Class
HTG-O	heating coil output	Heating Command	Heating Coil	Select Location Class
VAV	variable air volume box	Select Point Class	Variable Air Volume Box	Select Location Class
AHU	air handling unit	Select Point Class	Air Handling Unit	Select Location Class
HW-VLV	Heating Command	Heating Command	Select Equip Class	Select Location Class
DA-T	discharge air temperature sensor	Discharge Air Temperature Sensor	Select Equip Class	Select Location Class
OCC-CMD	Occupancy Command	Occupancy Command	Select Equip Class	Select Location Class
ZN-T	Zone Temperature Sensor	Zone Air Temperature Sensor	Select Equip Class	Select Location Class

Add/edit mappings as needed. Some abbreviations imply equipment, and some don’t

Home | Mapping | Models | Libraries | FDD Rules

BuildingMOTIF Model Commissioning Workflow Tutorial

This web application is a tutorial tool for the BuildingMOTIF software development kit (SDK). The goal of this tutorial is to demonstrate some key parts of the BuildingMOTIF model commissioning workflow.

1. Define a naming convention for the building's point labels, or upload an existing CSV
2. Upload point schedules create templates which will be used to create a metadata model of the building
3. Upload a point list to create a model

Proceed by clicking on each of the links below, in order. Start with Load Point Schedules.

We can now convert point schedules to templates

Setup

Define Naming Convention
Create Templates from Point Schedules

Setup (Advanced)

Configure Point Label Parser
Check Point Label Parse

Create Metadata Model

From Existing TTL file
From Existing Point Dump file

Model Validation

Create Model Manifest
Validate Most Recent Model (urn:buidling1)

Application Configuration

Generate FDD Rule Implementation Report

Database

Libraries
Models
Templates

If the UI is bugged, hit this button: Clear Demo State

Home | Mapping | Models | Libraries | FDD Rules

Point Schedule to Template

This tool converts a **point schedule** CSV file into a BuildingMOTIF template. Templates help generate semantic models by eliminating manual data entry when creating many similar points or equipment in the model. We generate the template in this step, and we will use the template later to help us build the m

Upload the point schedule CSV file below. The point schedule CSV file should have the following columns:

- **point**: contains the point abbreviation (e.g. `SA_T`)
- **description**: contains the point description (e.g. `Supply Air Temperature`)

explanatory image goes here

Now, click the **Generate** button below to generate the template. The template will be generated using an *offline large l* Brick classes. The template will be displayed below. If you are not satisfied with the template, you can regenerate it by

Template Name: my_vav_template

Target Brick Class

Variable Air Volume Box

Upload Point Schedule CSV: Browse... vav_point_list.csv

Generate

point	description
HW-VLV	Heating Command
DA-T	Discharge Air Temp Sensor
OCC-CMD	Occupancy Command
ZN-T	Zone Temperature Sensor
ZN-H	Zone Humidity Sensor
OCC-CLG-SP	Occupied Cooling Air Temp Setpoint
OCC-HTG-SP	Occupied Heating Air Temp Setpoint
UNOCC-CLG-SP	Unoccupied Cooling Air Temp Setpoint
UNOCC-HTG-SP	Unoccupied Heating Air Temp Setpoint
EFFCLG-SP	Effective Cooling Air Temp Setpoint
EFFHTG-SP	Effective Heating Air Temp Setpoint

Give the template a name, and upload the **Point Schedule CSV**

Pay attention to the column names!

Point Schedule to Template

This tool converts a **point schedule** CSV file into a BuildingMOTIF template. Templates help generate semantic models by eliminating manual data entry when creating many similar points or equipment in the model. We generate the template in this step, and we will use the template later to help us build the model itself.

Upload the point schedule CSV file below. The point schedule CSV file should have the following columns:

- **point**: contains the point abbreviation (e.g. `SA_T`)
- **description**: contains the point description (e.g. `Supply Air Temperature`)

explanatory image goes here

Now, click the **Generate** button below to generate the template. The template will be generated using an *offline large language embedding model* to map the point descriptions in the schedule to Brick classes. The template will be displayed below. If you are not satisfied with the template, you can regenerate it by clicking the **Regenerate** button.

Template Name: my_vav_template

Target Brick Class

Variable Air Volume Box

Upload Point Schedule CSV: Browse... vav_point_list.csv

Generate

Existing Templates

ahu-variant-1

ahu-variant-2

vav-with-damper

damper

vav-with-humidification

vav-with-reheat

reheat_valve

humidifier

my_vav_template

After a little bit (<1 minute) it should appear in this list.
Click for more details

Template: my_vav_template

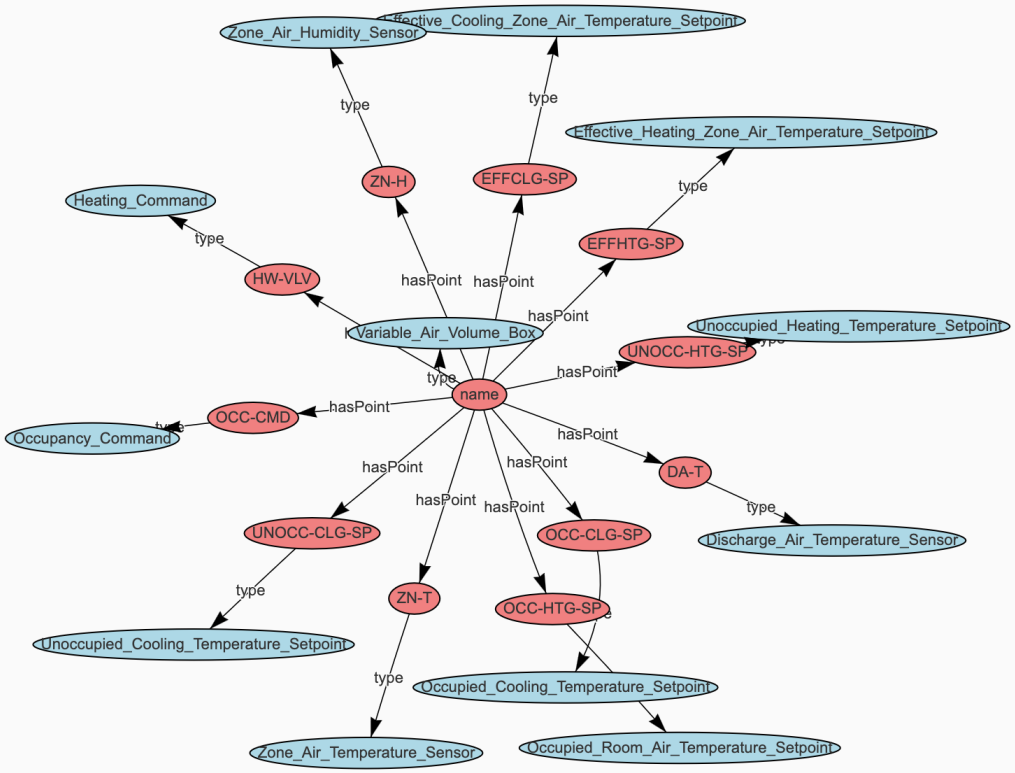
A template is a reusable function which automates creating parts of a metadata model. It has a set of parameters (red nodes, or those that begin with `urn:___param___`) that can be "bound" to values provided from BMS points, user input, or other sources. "Evaluating" a template creates a new graph that is a copy of the template, but with the parameters replaced with the bound values. The template is not modified by this process.

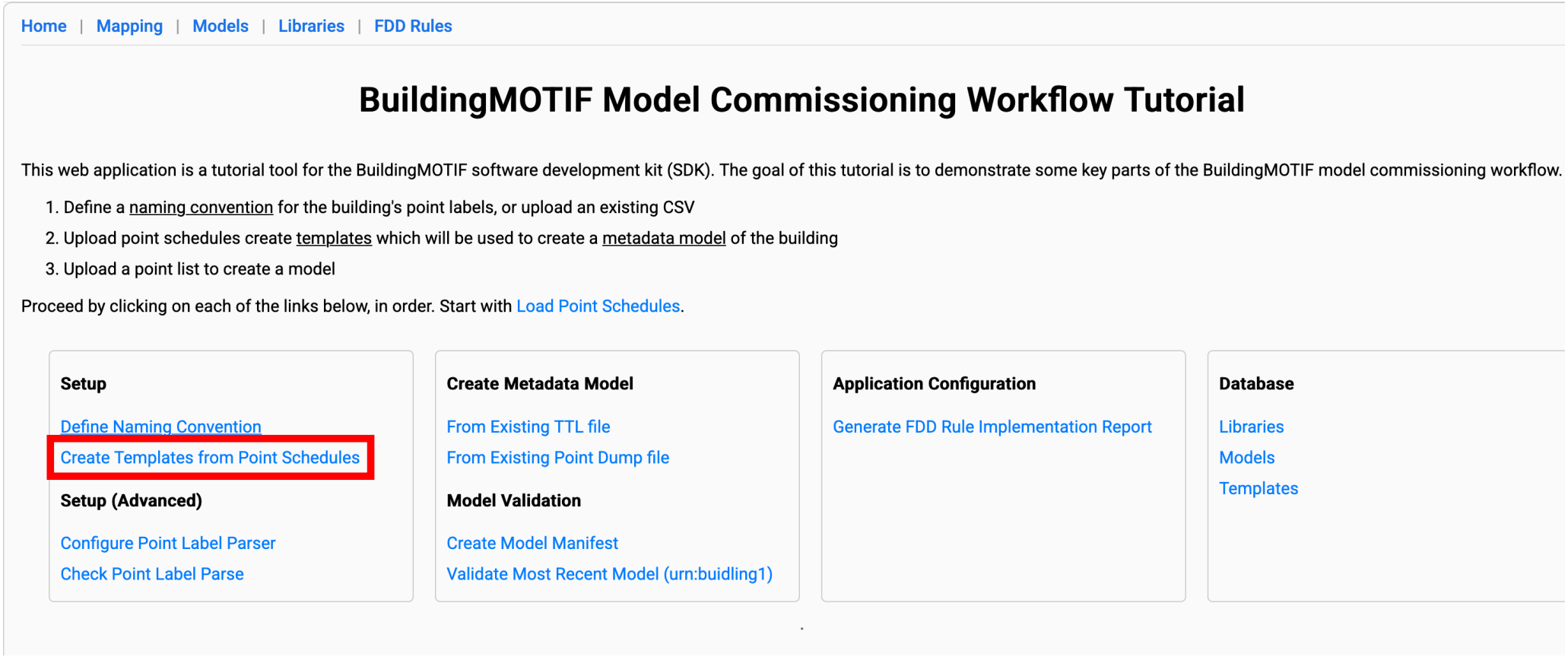
☐ Inline dependencies

Inspect the generated template for correctness.

You may need to adjust the ”mapping”/naming convention and try again.

Scroll down for a textual view of the same template





Repeat this process for all other point schedules you have.
Make sure to change the “target equipment” in the form.

Home | Mapping | Models | Libraries | FDD Rules

BuildingMOTIF Model Commissioning Workflow Tutorial

This web application is a tutorial tool for the BuildingMOTIF software development kit (SDK). The goal of this tutorial is to demonstrate some key parts of the BuildingMOTIF model commissioning workflow.

1. Define a naming convention for the building's point labels, or upload an existing CSV
2. Upload point schedules create templates which will be used to create a metadata model of the building
3. Upload a point list to create a model

Proceed by clicking on each of the links below, in order. Start with Load Point Schedules.

Setup

Define Naming Convention
Create Templates from Point Schedules

Setup (Advanced)

Configure Point Label Parser
Check Point Label Parse

Create Metadata Model

From Existing TTL file
From Existing Point Dump file

Model Validation

Create Model Manifest
Validate Most Recent Model (urn:buidling1)

Application Configuration

Generate FDD Rule Implementation Report

Database

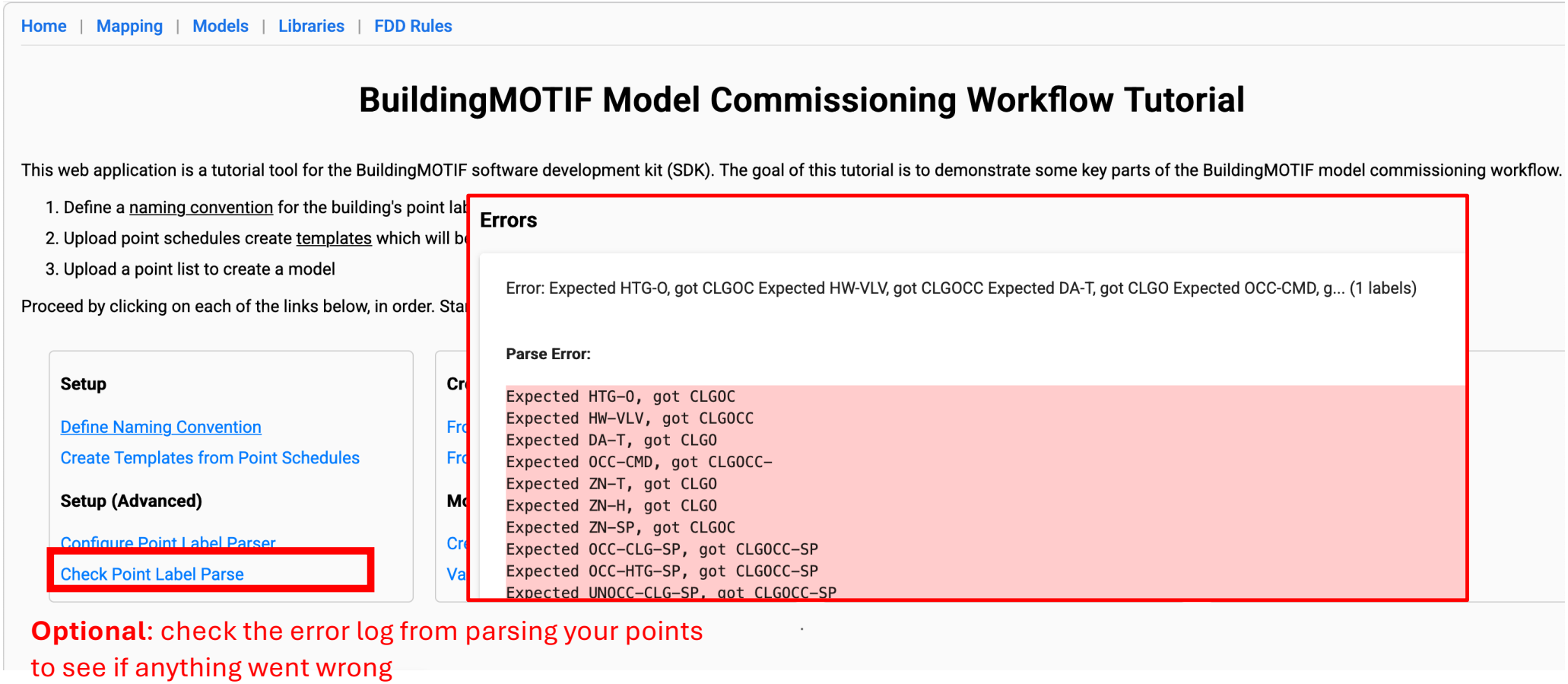
Libraries
Models
Templates

Optional: check the point label naming convention to help with parsing
(You do not need to do this for the demo)

```python

from buildingmotif.namespaces import BRICK
from buildingmotif.label_parsing.combinators import abbreviations, sequence, string, regex, wrap, constant
from buildingmotif.label_parsing.tokens import Delimiter, Identifier, Constant

# YOU HAVE ACCESS TO 'point_mappings' AND 'equipment_mappings' VARIABLES
# from the /mappings endpoint

short_equip_id = regex(r"\d+", Identifier)
building_id = sequence(
    constant(Constant(BRICK.Building)),
    regex(r"\d+", Identifier),
)

my_parser = sequence(
    building_id,
    string("_", Delimiter),
    equipment_mappings,
    string("_", Delimiter),
    short_equip_id,
    string("/", Delimiter),
    point_mappings
)

```

label
DEMO_VAV_01/DA-T
DEMO_VAV_01/DPR-O
DEMO_VAV_01/EFFCLG-SP
DEMO_VAV_01/EFFHTG-SP
DEMO_VAV_01/HTG-O
DEMO_VAV_01/SA-F
DEMO_VAV_01/SAFLOW-SP
DEMO_VAV_01/ZN-H
DEMO_VAV_01/ZN-T
DEMO_VAV_02/DA-T
DEMO_VAV_02/DPR-O
DEMO_VAV_02/EFFCLG-SP
DEMO_VAV_02/EFFHTG-SP
DEMO_VAV_02/HTG-O
DEMO_VAV_02/SA-F
DEMO_VAV_02/SAFLOW-SP
DEMO_VAV_02/ZN-H

- Check that the “my_parser” naming convention matches the naming convention of your point labels
- See https://gtf.fyi/files/2025-ASHRAE-Brownfield-Fierro.pdf for some more detailed slides on how this parser works
- Click here for docs on the parser module



Optional: check the error log from parsing your points to see if anything went wrong

- Typos
- Missing abbreviations

Home | Mapping | Models | Libraries | FDD Rules

BuildingMOTIF Model Commissioning Workflow Tutorial

This web application is a tutorial tool for the BuildingMOTIF software development kit (SDK). The goal of this tutorial is to demonstrate some key parts of the BuildingMOTIF model commissioning workflow.

1. Define a naming convention for the building's point labels, or upload an existing CSV
2. Upload point schedules create templates which will be used to create a metadata model of the building
3. Upload a point list to create a model

Proceed by clicking on each of the links below, in order. Start with Load Point Schedules.

Setup

Define Naming Convention
Create Templates from Point Schedules

Setup (Advanced)

Configure Point Label Parser
Check Point Label Parse

Create Metadata Model

From Existing TTL file
From Existing Point Dump file

Model Validation

Create Model Manifest
Validate Most Recent Model (urn:buidling1)

Application Configuration

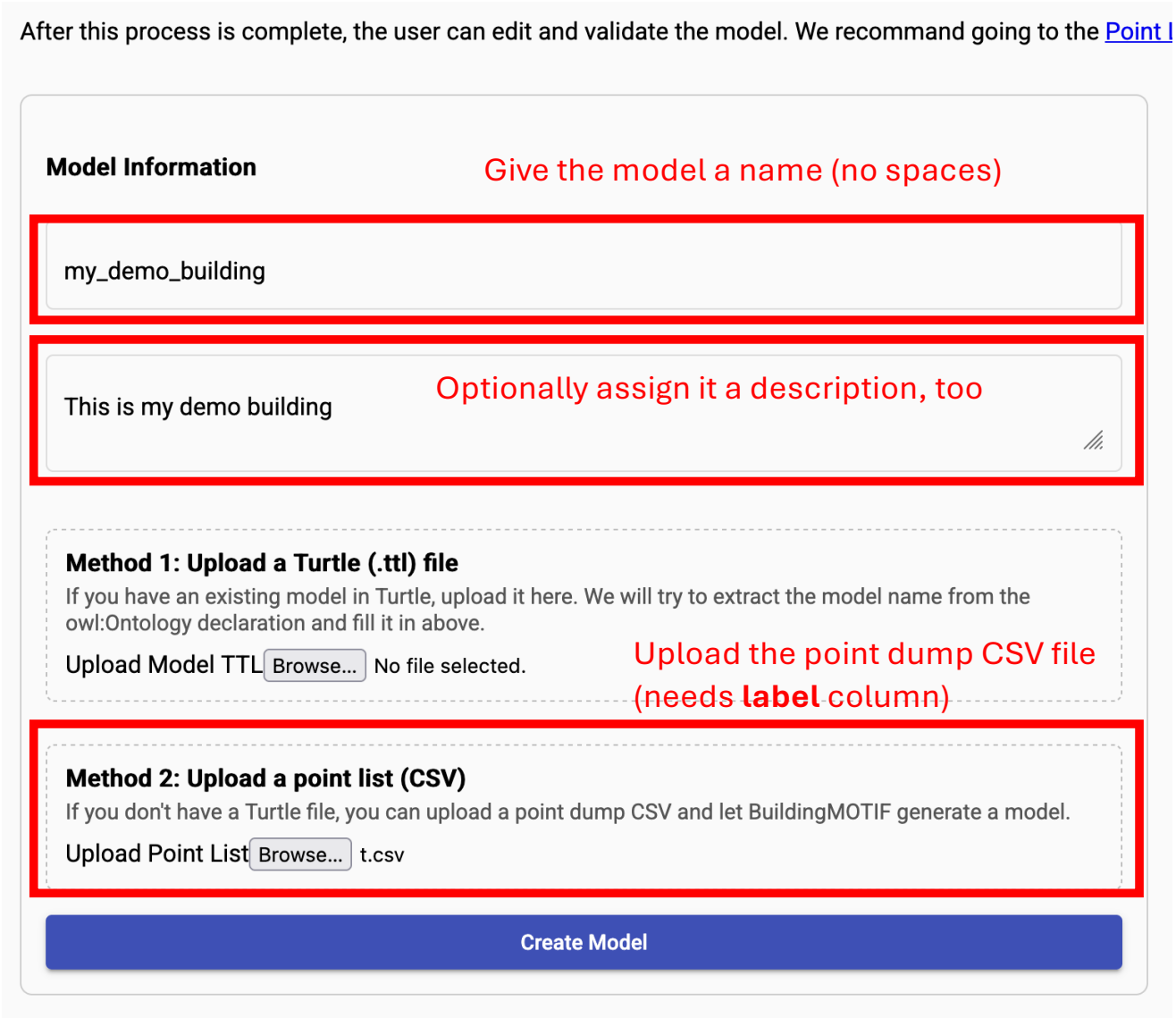
Generate FDD Rule Implementation Report

Database

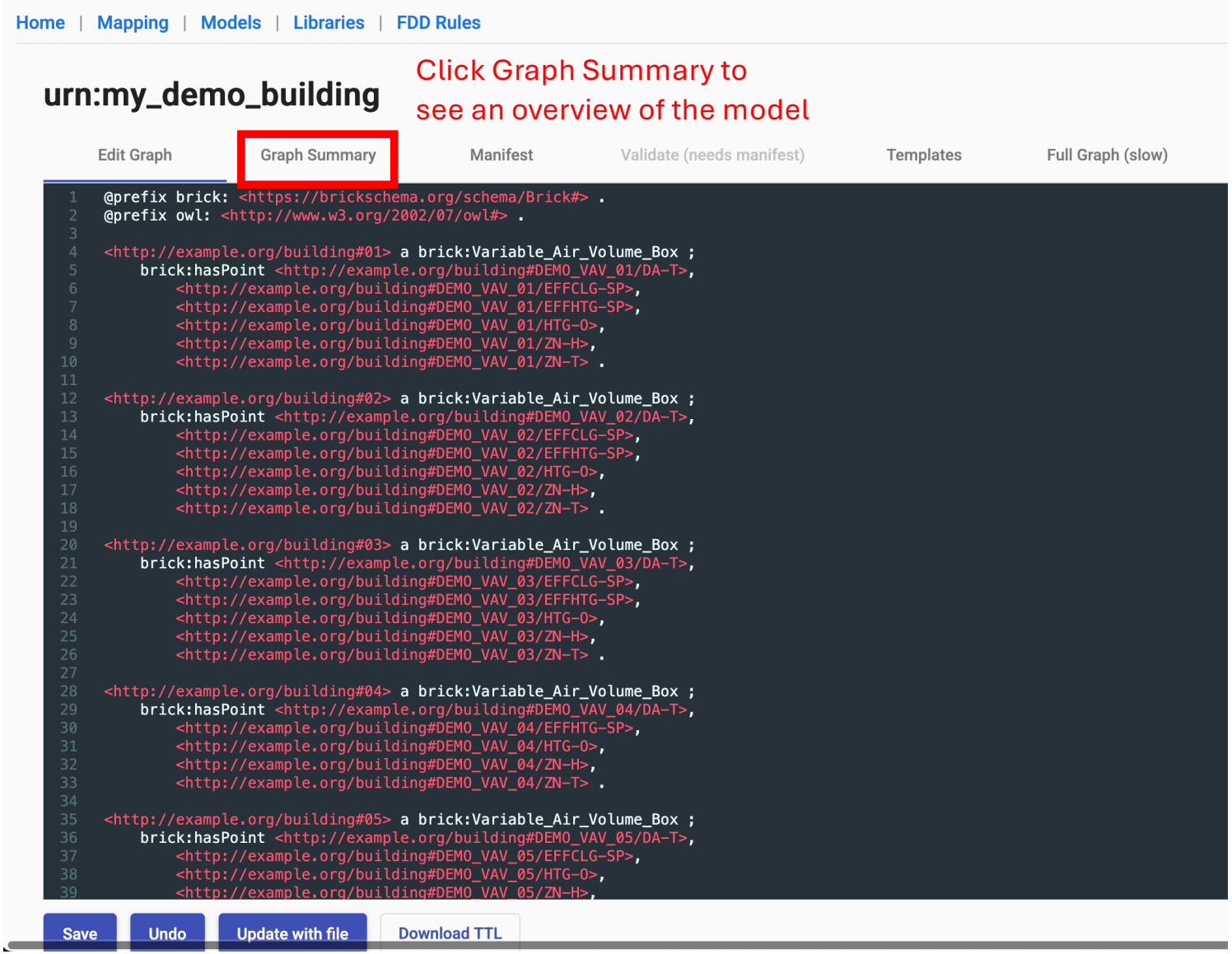
Libraries
Models
Templates

Now, we can create a metadata model!

After this process is complete, the user can edit and validate the model. We recommand going to the Point I

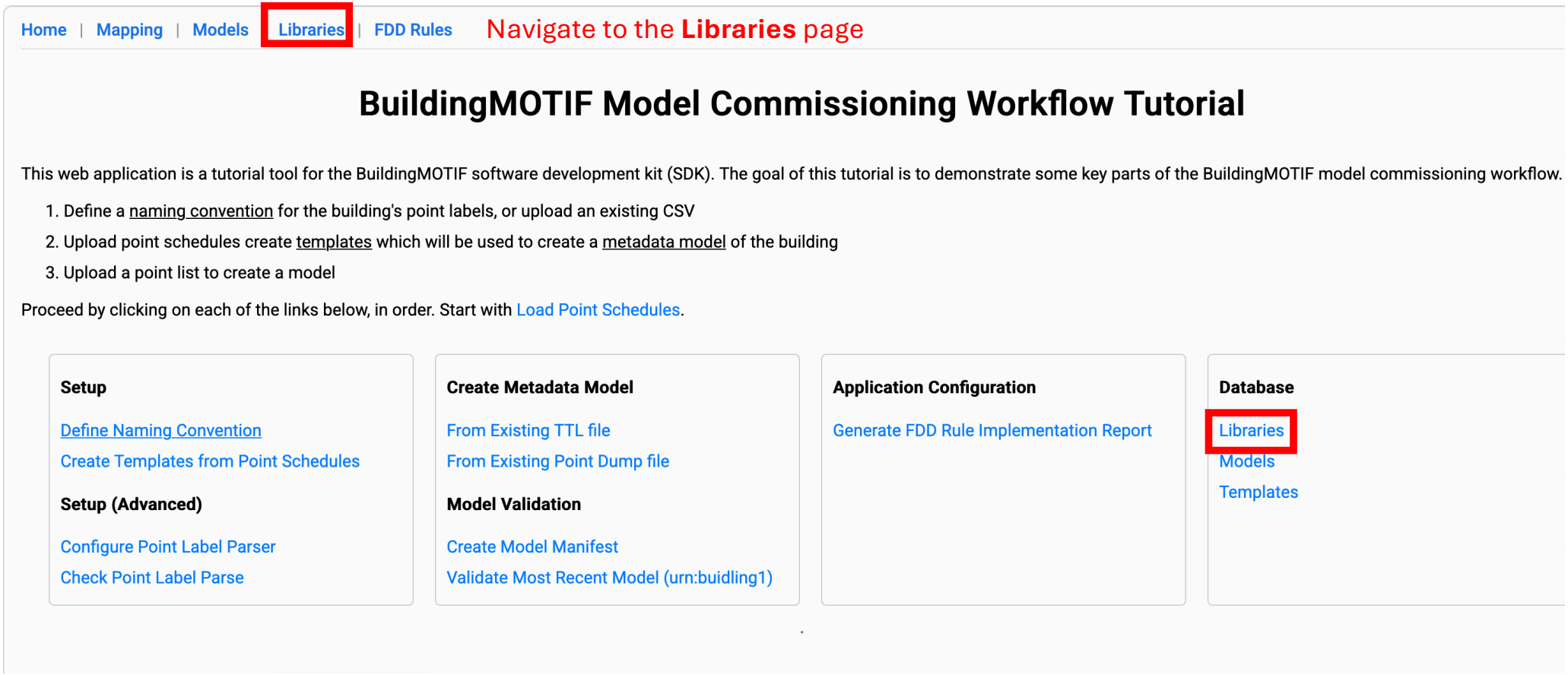


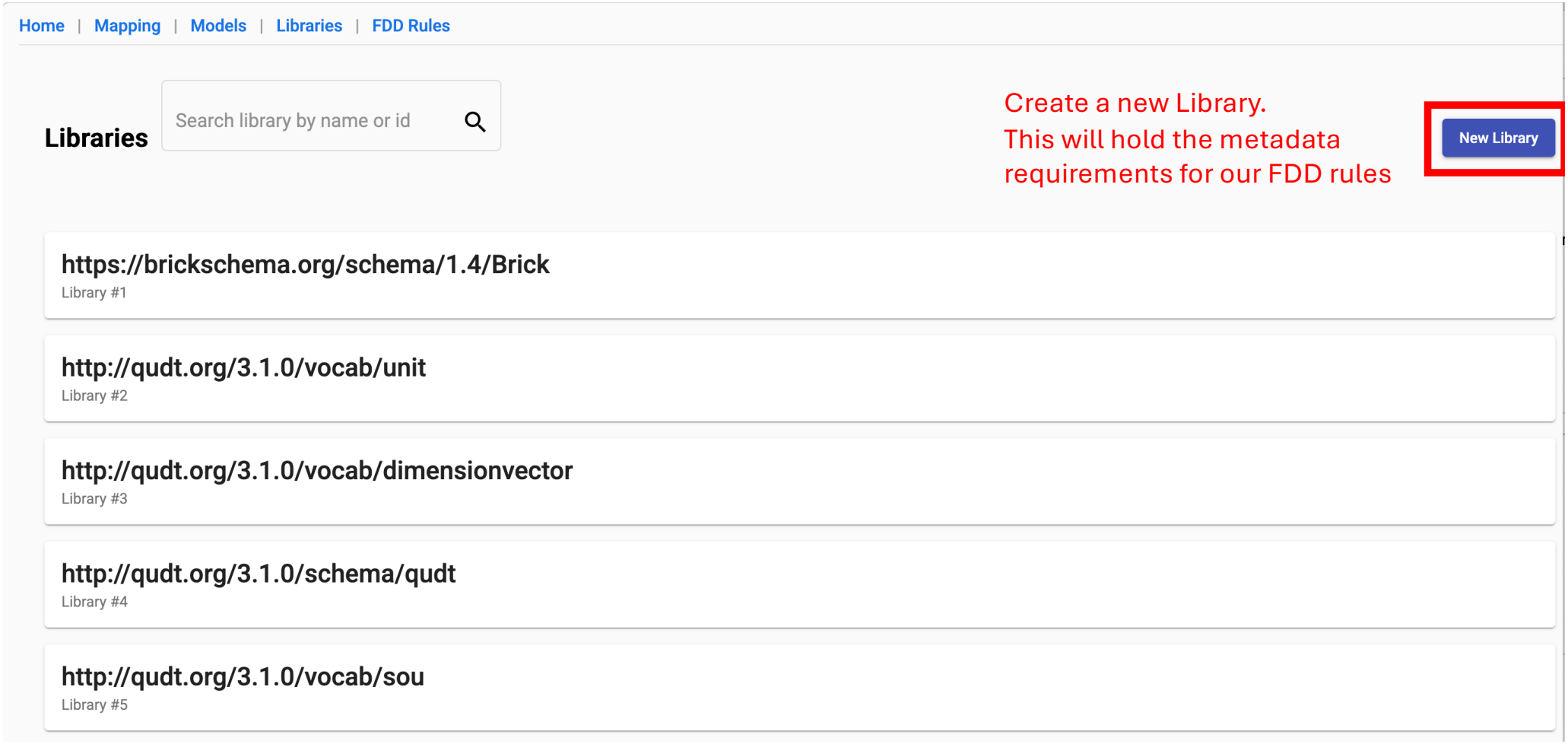
label
DEMO_VAV_01/DA-T
DEMO_VAV_01/DPR-O
DEMO_VAV_01/EFFCLG-SP
DEMO_VAV_01/EFFHTG-SP
DEMO_VAV_01/HTG-O
DEMO_VAV_01/SA-F
DEMO_VAV_01/SAFLOW-SP
DEMO_VAV_01/ZN-H
DEMO_VAV_01/ZN-T
DEMO_VAV_02/DA-T
DEMO_VAV_02/DPR-O
DEMO_VAV_02/EFFCLG-SP
DEMO_VAV_02/EFFHTG-SP
DEMO_VAV_02/HTG-O
DEMO_VAV_02/SA-F
DEMO_VAV_02/SAFLOW-SP
DEMO_VAV_02/ZN-H



After a couple of minutes, the initial model will be ready!

Now that the initial model is done, let’s start validating


Home | Mapping | Models | Libraries | FDD Rules
Navigate to the Libraries page
BuildingMOTIF Model Commissioning Workflow Tutorial
This web application is a tutorial tool for the BuildingMOTIF software development kit (SDK). The goal of this tutorial is to demonstrate some key parts of the BuildingMOTIF model commissioning workflow.
1. Define a naming convention for the building's point labels, or upload an existing CSV
2. Upload point schedules create templates which will be used to create a metadata model of the building
3. Upload a point list to create a model
Proceed by clicking on each of the links below, in order. Start with Load Point Schedules.
Setup
Define Naming Convention
Create Templates from Point Schedules
Setup (Advanced)
Configure Point Label Parser
Check Point Label Parse
Create Metadata Model
From Existing TTL file
From Existing Point Dump file
Model Validation
Create Model Manifest
Validate Most Recent Model (urn:buidling1)
Application Configuration
Generate FDD Rule Implementation Report
Database
Libraries
Models
Templates


Home | Mapping | Models | Libraries | FDD Rules
Libraries
Search library by name or id
Create a new Library.
This will hold the metadata
requirements for our FDD rules
New Library
https://brickschema.org/schema/1.4/Brick
Library #1
http://qudt.org/3.1.0/vocab/unit
Library #2
http://qudt.org/3.1.0/vocab/dimensionvector
Library #3
http://qudt.org/3.1.0/schema/qudt
Library #4
http://qudt.org/3.1.0/vocab/sou
Library #5

Create New Library

From TTL

Upload a Turtle (.ttl) file describing your library. We'll try to infer a name automatically.

Choose TTL File

Library Name

Create Library

From Rules JSON

Upload a rules.json file; the backend will transform it into a SHACL library.

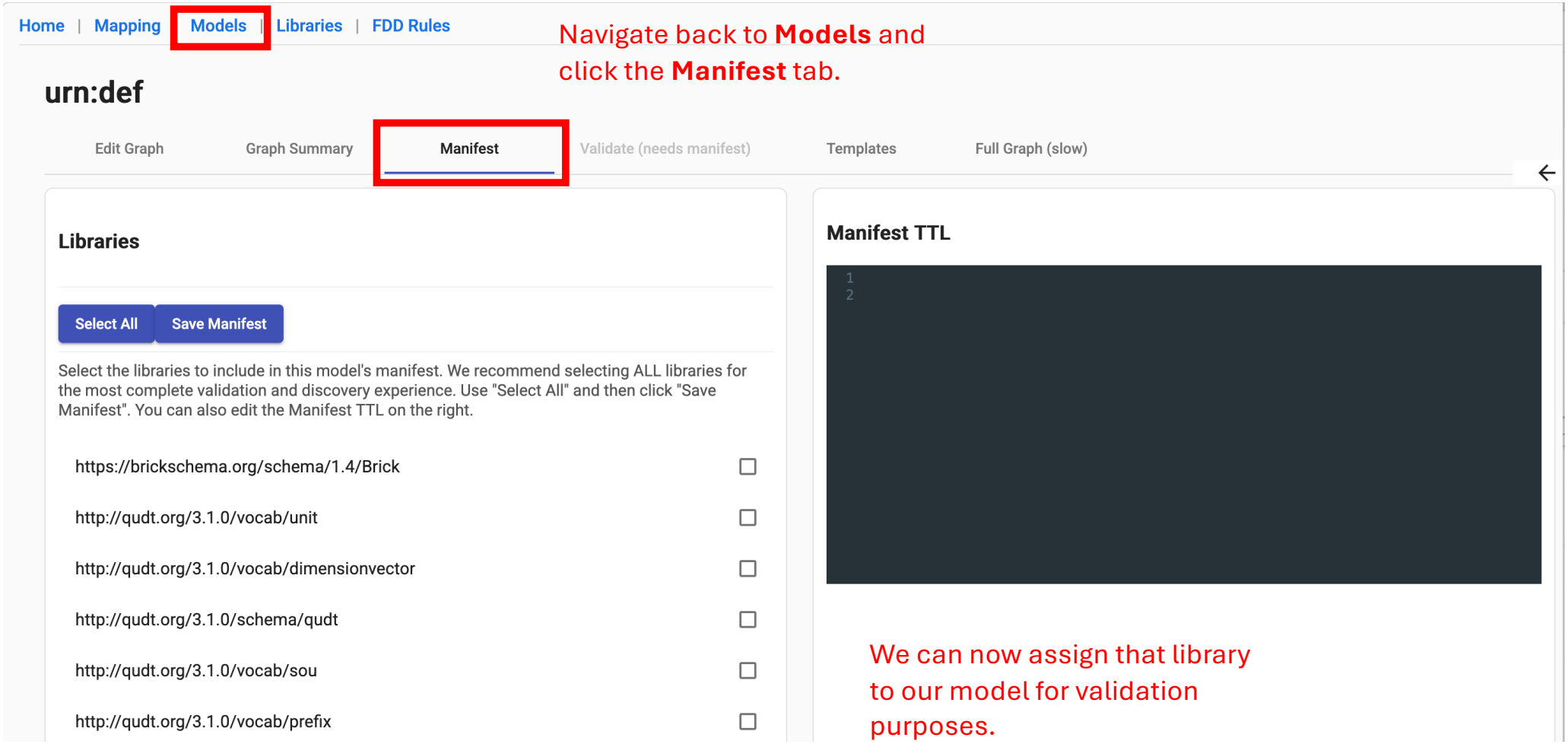
Choose Rules JSON GL36_FCU.json

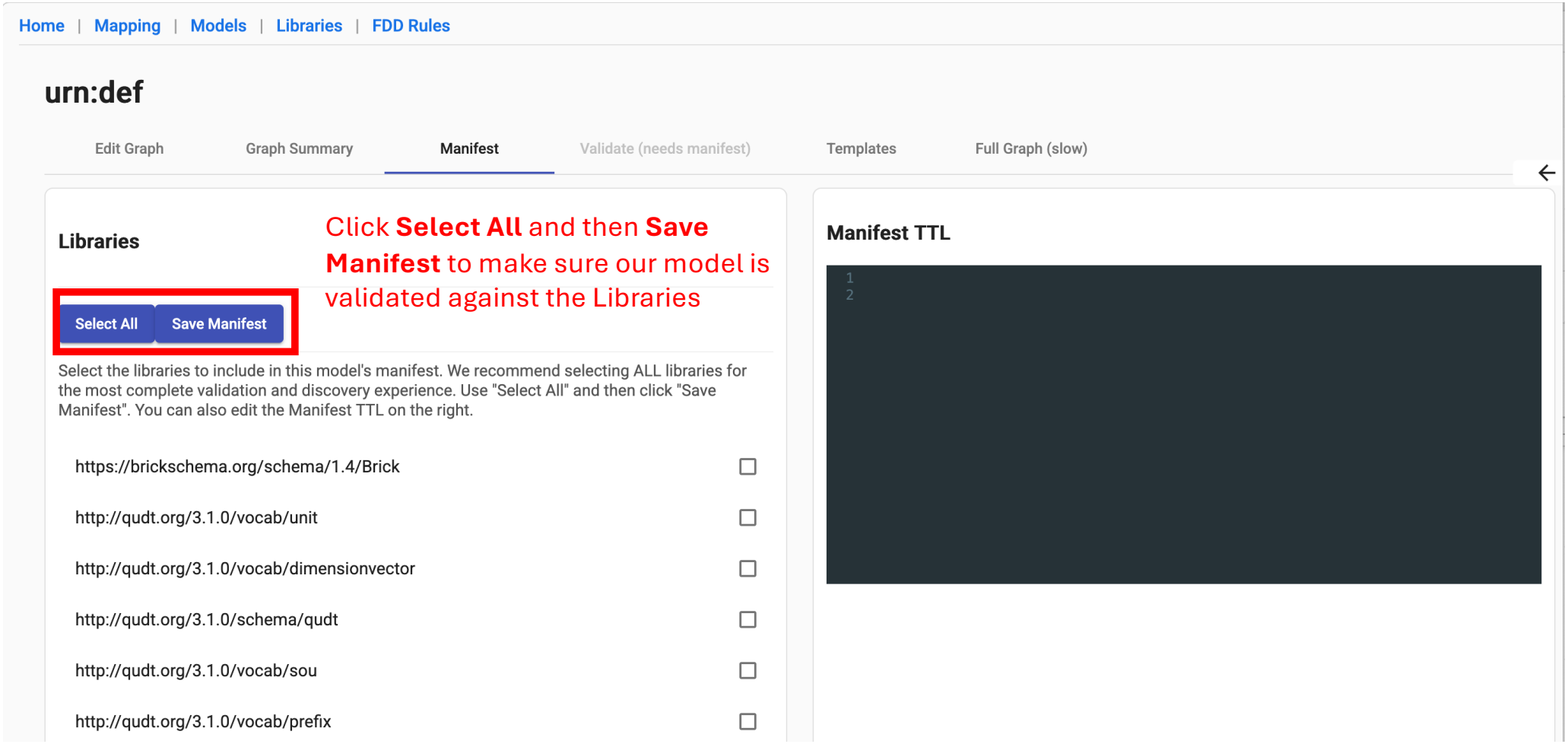
Library Name (optional)

GL36_FCU

Upload the rules JSON file

Create Library


Home | Mapping | Models | Libraries | FDD Rules
Navigate back to **Models** and click the **Manifest** tab.
urn:def
Edit Graph
Graph Summary
Manifest
Validate (needs manifest)
Templates
Full Graph (slow)
Libraries
Select All
Save Manifest
Select the libraries to include in this model's manifest. We recommend selecting ALL libraries for the most complete validation and discovery experience. Use "Select All" and then click "Save Manifest". You can also edit the Manifest TTL on the right.
https://brickschema.org/schema/1.4/Brick
http://qudt.org/3.1.0/vocab/unit
http://qudt.org/3.1.0/vocab/dimensionvector
http://qudt.org/3.1.0/schema/qudt
http://qudt.org/3.1.0/vocab/sou
http://qudt.org/3.1.0/vocab/prefix
Manifest TTL
1
2
We can now assign that library to our model for validation purposes.


Home | Mapping | Models | Libraries | FDD Rules
urn:def
Edit Graph
Graph Summary
Manifest
Validate (needs manifest)
Templates
Full Graph (slow)
Libraries
Click Select All and then Save Manifest to make sure our model is validated against the Libraries
Select All
Save Manifest
Select the libraries to include in this model's manifest. We recommend selecting ALL libraries for the most complete validation and discovery experience. Use "Select All" and then click "Save Manifest". You can also edit the Manifest TTL on the right.
https://brickschema.org/schema/1.4/Brick
http://qudt.org/3.1.0/vocab/unit
http://qudt.org/3.1.0/vocab/dimensionvector
http://qudt.org/3.1.0/schema/qudt
http://qudt.org/3.1.0/vocab/sou
http://qudt.org/3.1.0/vocab/prefix
Manifest TTL
1
2

urn:my_demo_building

Edit Graph | Graph Summary | Manifest | Validate (needs manifest) | Templates | Full Graph (slow)

Libraries

Select All | Save Manifest

Select the libraries to include in this model's manifest. We recommend selecting ALL libraries for the most complete validation and discovery experience. Use "Select All" and then click "Save Manifest". You can also edit the Manifest TTL on the right.

- https://brickschema.org/schema/1.4/Brick ☑
- http://qudt.org/3.1.0/schema/extensions/functions ☑
- http://qudt.org/3.1.0/vocab/quantitykind ☑
- http://qudt.org/3.1.0/vocab/sou ☑
- http://qudt.org/3.1.0/schema/facade/qudt ☑
- http://qudt.org/3.1.0/schema/qudt ☑
- http://qudt.org/3.1.0/vocab/prefix ☑

Manifest TTL

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<urn:my_demo_building> owl:imports <http://qudt.org/3.1.0/schema/extensions/functions>,
        <http://qudt.org/3.1.0/schema/facade/qudt>,
        <http://qudt.org/3.1.0/schema/qudt>,
        <http://qudt.org/3.1.0/vocab/dimensionvector>,
        <http://qudt.org/3.1.0/vocab/prefix>,
        <http://qudt.org/3.1.0/vocab/quantitykind>,
        <http://qudt.org/3.1.0/vocab/sou>,
        <http://qudt.org/3.1.0/vocab/unit>,
        <https://brickschema.org/schema/1.4/Brick>,
        <https://nrel.gov/BuildingMOTIF/constraints>,
        <urn:GL36_FCU.json>,
        <urn:asbuilt-lib> .


```

We can see all of the libraries used to validate our model.

NOW click the “Validate” tab and wait a few minutes for validation to complete.

This only needs to run once every time you change the model

urn:my_demo_building

Edit Graph | Graph Summary | Manifest | Validate | Templates | Full Graph (slow)

Re-run Validation

Valid? 0

Validation Result

http://example.org/building#01

- http://example.org/building#01 expected at least 1 instance(s) of brick:Occupancy_Command on path brick:hasPoint
- http://example.org/building#01 expected at least 1 instance(s) of brick:Occupied_Room_Air_Temperature_Setpoint on path brick:hasPoint
- http://example.org/building#01 expected at least 1 instance(s) of brick:Unoccupied_Cooling_Temperature_Setpoint on path brick:hasPoint
- http://example.org/building#01 expected at least 1 instance(s) of brick:Unoccupied_Heating_Temperature_Setpoint on path brick:hasPoint
- http://example.org/building#01 expected at least 1 instance(s) of brick:Occupied_Cooling_Temperature_Setpoint on path brick:hasPoint

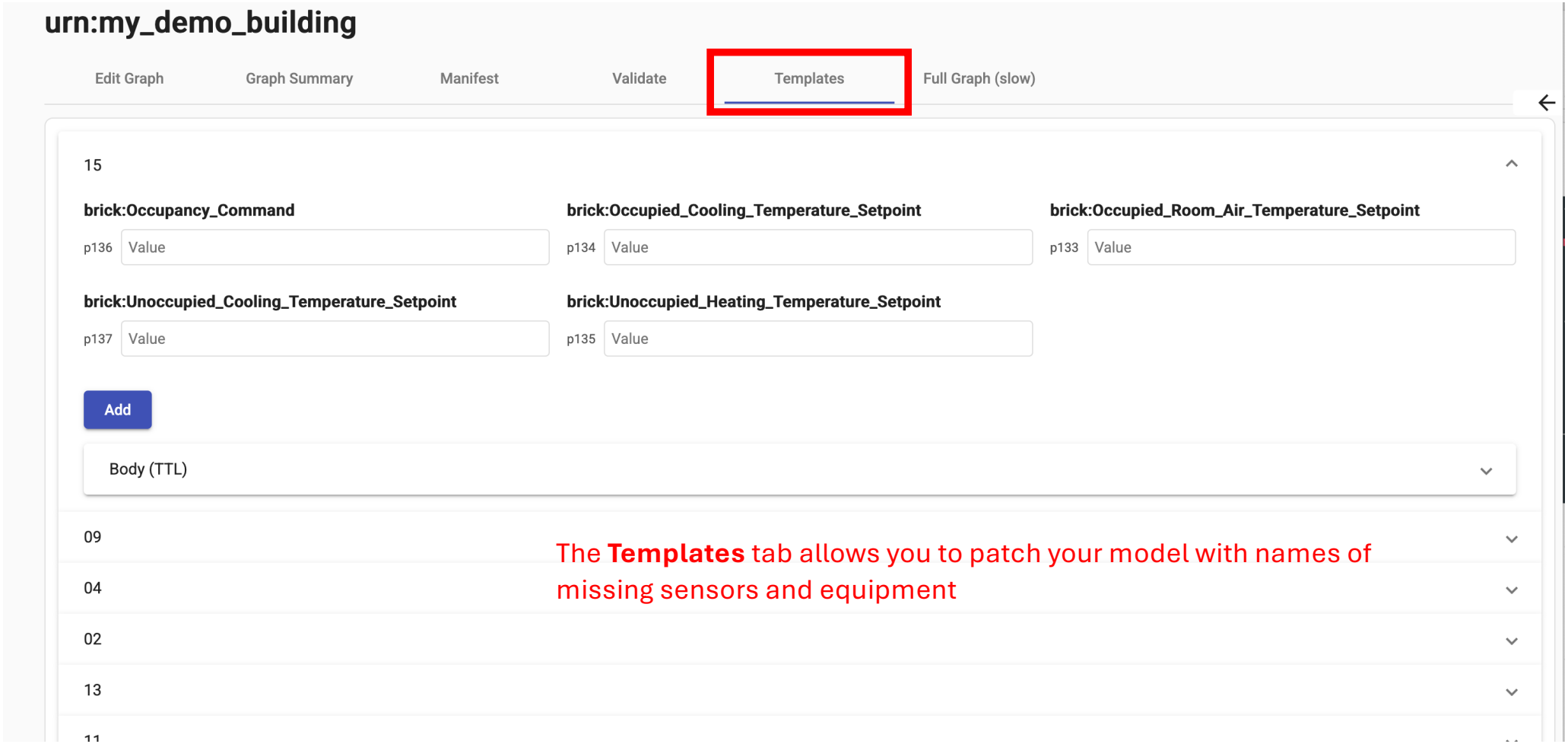
http://example.org/building#02

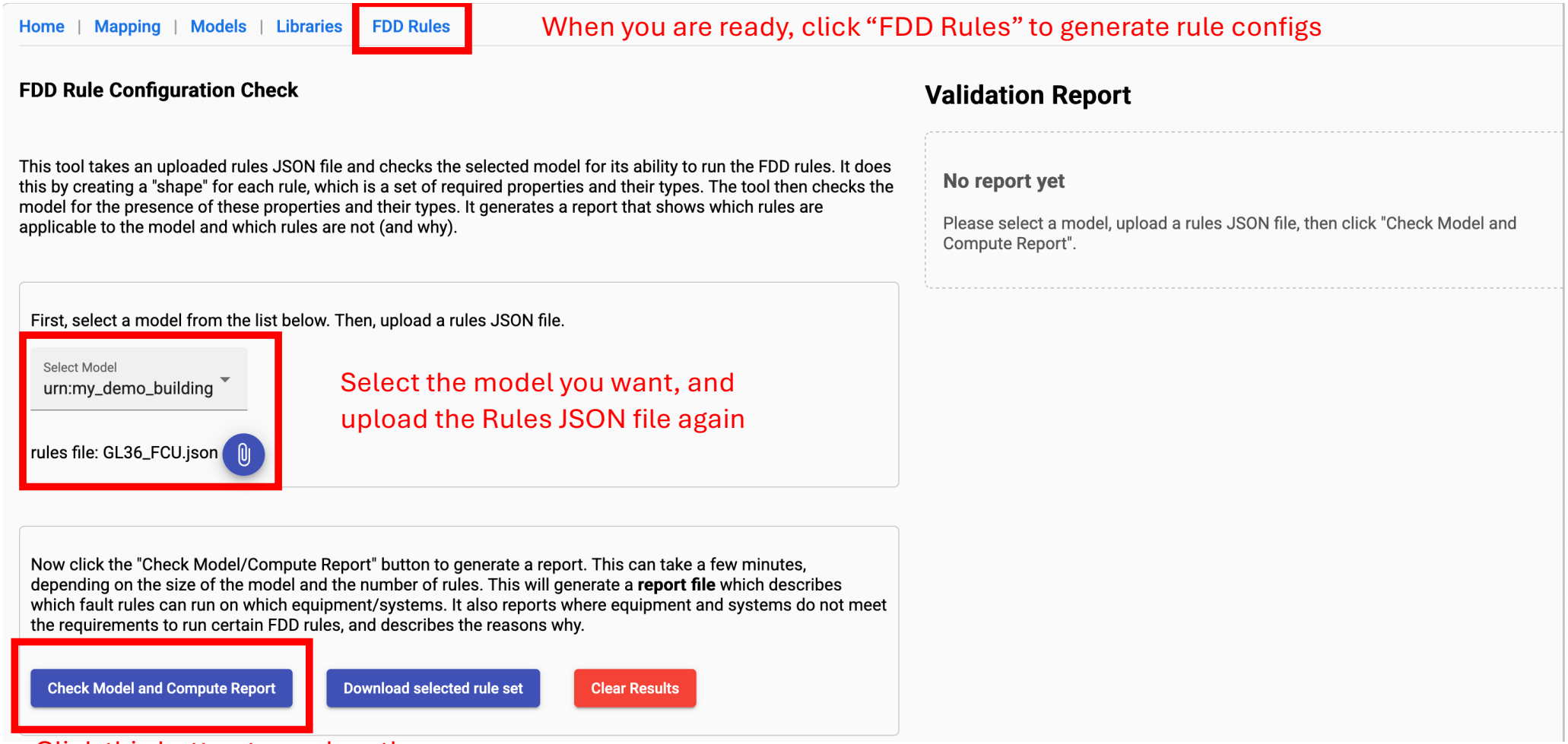
http://example.org/building#03

http://example.org/building#04

http://example.org/building#05

The validation results list what is “wrong”/missing about each equipment or point in the model


urn:my_demo_building
Edit Graph
Graph Summary
Manifest
Validate
Templates
Full Graph (slow)
15
brick:Occupancy_Command
p136
Value
brick:Occupied_Cooling_Temperature_Setpoint
p134
Value
brick:Occupied_Room_Air_Temperature_Setpoint
p133
Value
brick:Unoccupied_Cooling_Temperature_Setpoint
p137
Value
brick:Unoccupied_Heating_Temperature_Setpoint
p135
Value
Add
Body (TTL)
09
04
02
13
11
The Templates tab allows you to patch your model with names of missing sensors and equipment


Home | Mapping | Models | Libraries | FDD Rules
When you are ready, click “FDD Rules” to generate rule configs
FDD Rule Configuration Check
This tool takes an uploaded rules JSON file and checks the selected model for its ability to run the FDD rules. It does this by creating a "shape" for each rule, which is a set of required properties and their types. The tool then checks the model for the presence of these properties and their types. It generates a report that shows which rules are applicable to the model and which rules are not (and why).
First, select a model from the list below. Then, upload a rules JSON file.
Select Model
urn:my_demo_building
rules file: GL36_FCU.json
Select the model you want, and upload the Rules JSON file again
Now click the "Check Model/Compute Report" button to generate a report. This can take a few minutes, depending on the size of the model and the number of rules. This will generate a report file which describes which fault rules can run on which equipment/systems. It also reports where equipment and systems do not meet the requirements to run certain FDD rules, and describes the reasons why.
Check Model and Compute Report
Download selected rule set
Clear Results
Click this button to analyze the model for sufficient information to configure the FDD rules. This can take a few minutes
Validation Report
No report yet
Please select a model, upload a rules JSON file, then click "Check Model and Compute Report".

FDD Rule Configuration Check

This tool takes an uploaded rules JSON file and checks the selected model for its ability to run the FDD rules. It does this by creating a "shape" for each rule, which is a set of required properties and their types. The tool then checks the model for the presence of these properties and their types. It generates a report that shows which rules are applicable to the model and which rules are not (and why).

First, select a model from the list below. Then, upload a rules JSON file.

Select Model
urn:my_demo_building

rules file: GL36_FCU.json

Now click the "Check Model/Compute Report" button to generate a report. This can take a few minutes, depending on the size of the model and the number of rules. This will generate a **report file** which describes which fault rules can run on which equipment/systems. It also reports where equipment and systems do not meet the requirements to run certain FDD rules, and describes the reasons why.

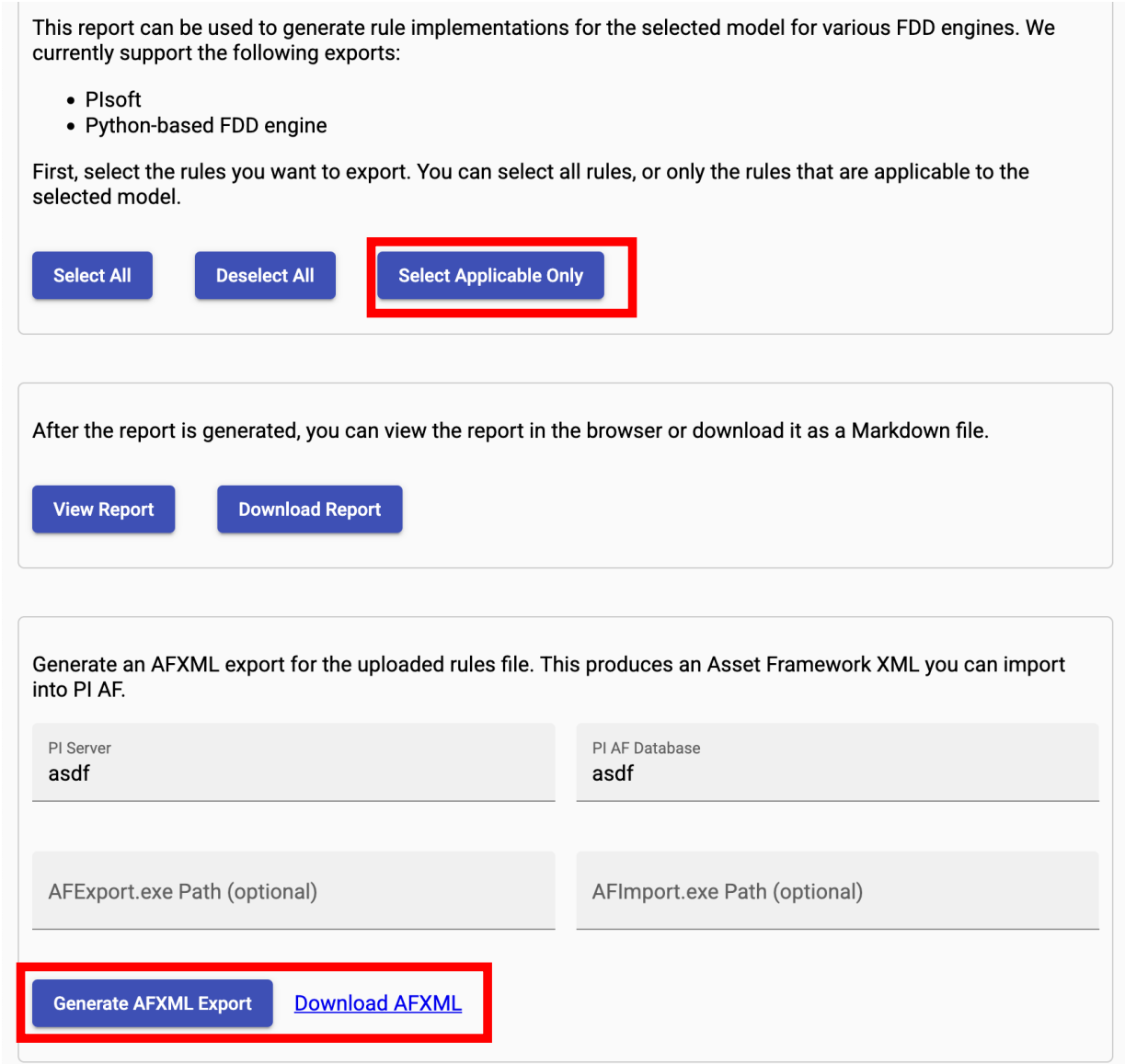
Check Model and Compute Report | Download selected rule set | Clear Results

Validation Report

By Rule / Application | By Asset

- [x] http://example.org/building#G36FCU_SAT_POINTS
- [ ] http://example.org/building#Guideline_36_-_All_supply_air_temperature_control_points_and_sensors_are_available_for_AFDD
- [ ] http://example.org/building#var7HPR
- [ ] http://example.org/building#var882F
- [ ] http://example.org/building#varE5TY
- [ ] http://example.org/building#varIVKD
- [ ] urn:well-known/1c13e0f7
- [ ] urn:well-known/1ebad89e
- [ ] urn:well-known/3ee557b2
- [ ] urn:well-known/3fc0662b
- [ ] urn:well-known/4c34519e
- [ ] urn:well-known/50c9d274

This report tells us which FDD rules we can/cannot successfully configure



Click **Select Applicable Only** to select only the rule/equip combos that can be correctly configured

Fill in information about your PI Server and PI AF Database

Then Generate and Download the AFXML to upload into your PI Server