Google

Module 1 | **Lesson 5**

Digital Buildings Ontology (DBO)

Before you get started

This onboarding deck has interactive features and activities that enable a self-guided learning experience.
To help you get started, here are two tips for viewing and navigating through the deck.

① **View this deck in presentation mode.**

- To enter presentation mode, you can either:
  - Click the **Present** or **Slideshow** button in the top-right corner of this page.
  - Press **Ctrl+F5** (Windows), **Cmd+Enter** (macOS), or **Ctrl+Search+5** (Chrome OS) on your keyboard.
- To exit presentation mode, press the **Esc** key on your keyboard.

② **Navigate by clicking the buttons and links.**

- Click the **Back** or **Next** buttons to go backward or forward in the deck. Moving forward, you'll find them in the bottom corners of every slide.
- Click blue text to go to another slide in this deck or open a new page in your browser.
- For the best learning experience, using your keyboard or mouse wheel to navigate is discouraged.

Ready to get started? Let's go!

Conceptual model revisited

Here's another look at the DBO conceptual model from Lesson 2.

In this lesson, you'll explore one modeling concept from the abstract model. Remember, abstract modeling concepts are used to describe the properties of an entity. Abstract concepts include:

- Subfields
- Fields
- States and multi-states
- Entity types

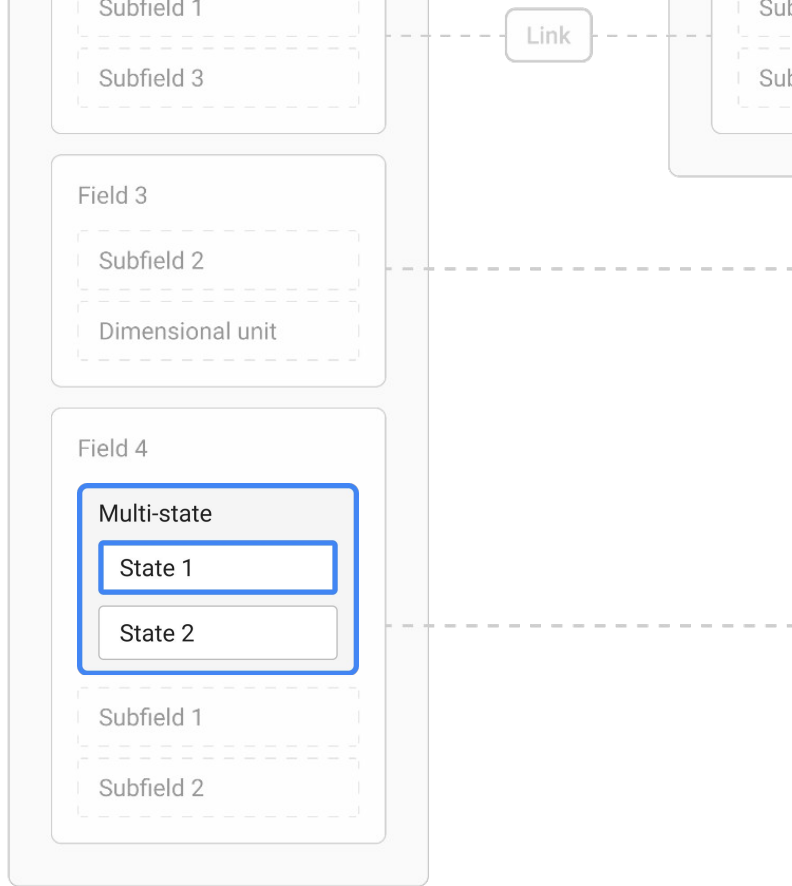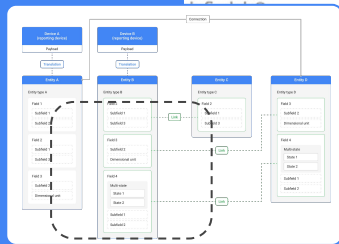**Do you see these concepts in the diagram?**
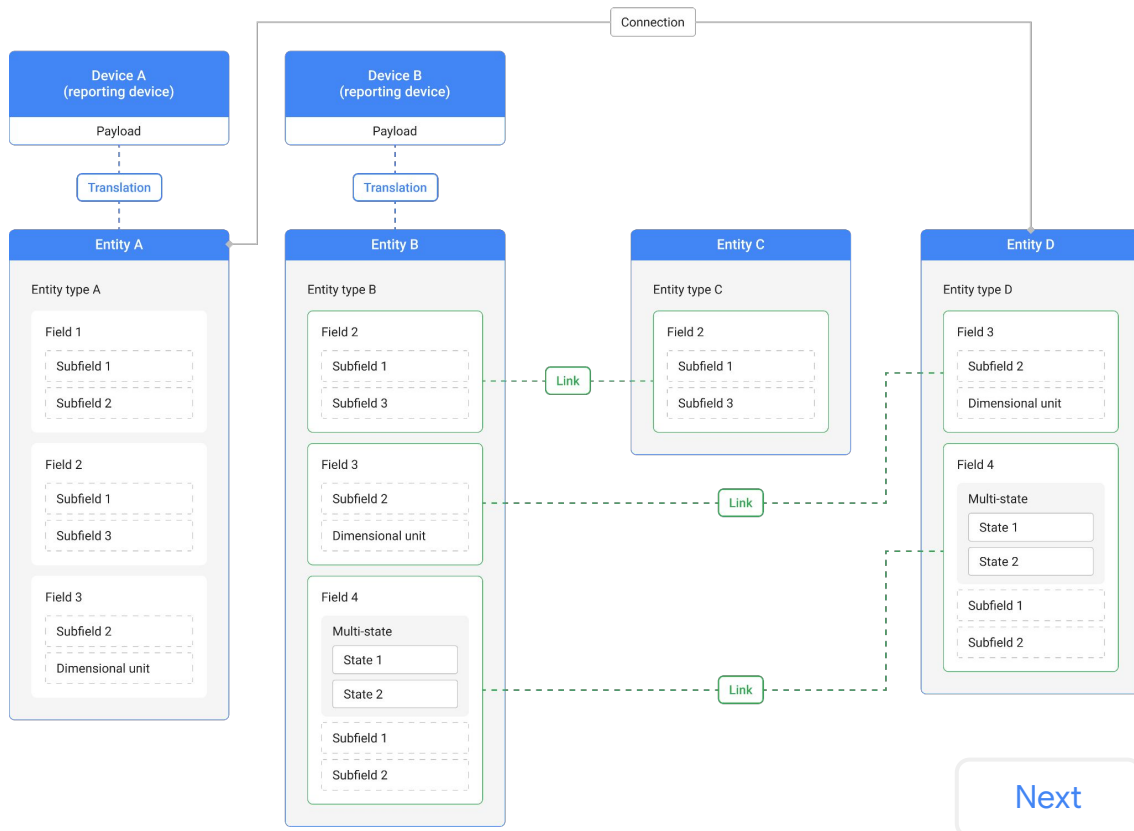
Lesson 5

States and multi-states

What you'll learn about:

- States
- Multi-states
- State and multi-state constraints

By the end of this lesson, you'll be able to:

- Describe the concepts of a state and multi-state.
- Identify a multi-state in source code.
- Understand the rules and constraints for multi-states.

Back

Next

What's a state?

A state is a single word or string that describes a device's condition, status, setting, position, mode, etc.

Similar to subfields, each state also consists of a specific, human-readable definition.

Examples

ON: Powered on.

OFF: Powered off.

**LOW:** Low speed or output setting.

MEDIUM: Medium speed or output setting.

HIGH: High speed or output setting.

See states.yaml in the Digital Buildings Project GitHub repo for a list of all available states.

Link

Subfield 3

Field 3

Subfield 2

Dimensional unit

Field 4

Multi-state

State 1

State 2

Subfield 1

Subfield 2

Back

Note: Remember, a field is a semantic definition of an abstract concept. Abstract concepts are associated with an entity, which is the concrete instance of the device to be modeled.

Next
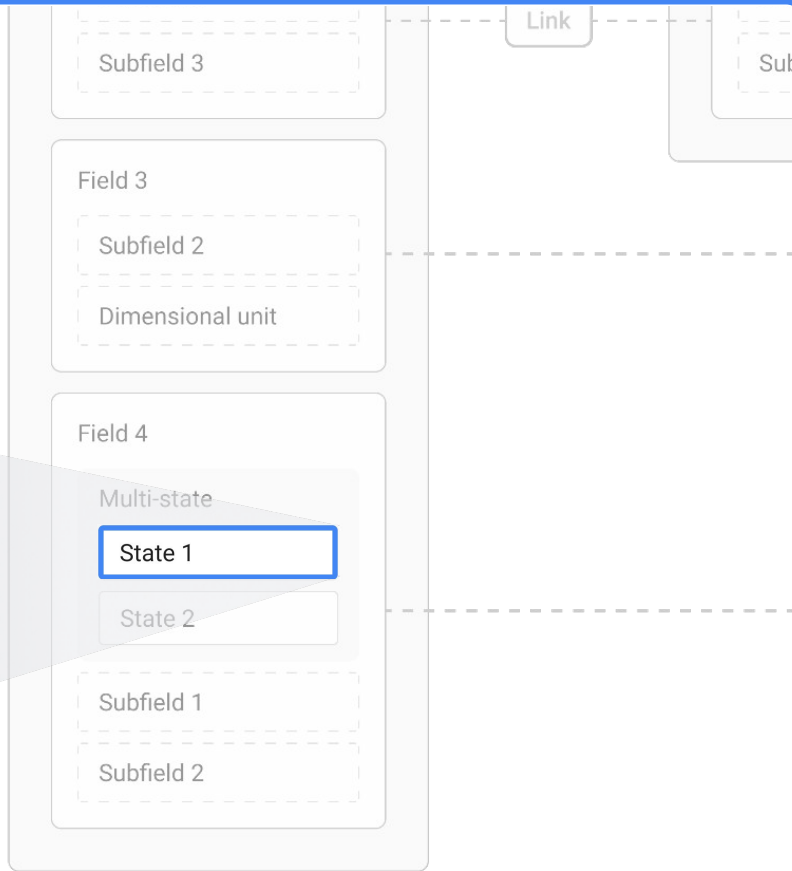
What's a multi-state?

A multi-state is any field that sends states.

Each multi-state field consists of a list of allowed states that can describe a device's condition, status, setting, position, mode, etc. Multi-states are defined on a field-by-field basis.

Examples

```
supply_fan_failed_alarm:
  - ACTIVE
  - INACTIVE

battery_charge_status:
  - CHARGING
  - DISCHARGING
  - STANDBY
```

```
control_mode:
  - AUTO
  - MANUAL
  - OFF

schedule_run_command:
  - ON
  - OFF
```

Subfield 3

Link

Sub

Field 3

Subfield 2

Dimensional unit

Field 4

Multi-state

State 1

State 2

Subfield 1

Subfield 2

Back

Next

Google

State and multi-state constraints

Remember the following constraints when working with states and multi-states.

State constraints

States should:

- Always have a specific definition that's the same wherever it's used.
- Normally apply globally wherever possible.
- Rarely be defined only within a namespace.

Multi-state constraints

A field is considered a multi-state field if one of the following is true:

- The point type subfield is `alarm`, `status`, or `mode`.
- The point type subfield is `command`, and it doesn't have a measurement subfield.

Examples

```
supply_fan_failed_alarm:          control_mode:
 - ACTIVE                           - AUTO
 - INACTIVE                         - MANUAL
                                    - OFF

battery_charge_status:
 - CHARGING                        schedule_run_command:
 - DISCHARGING                      - ON
 - STANDBY                          - OFF
```

Back

Next

Lesson 5

Knowledge check

Let's take a moment to reflect on what you've learned so far.

- The next slides will have questions about the concepts that were introduced in this lesson.

- Review each question and select the correct response.

If there are more than two answer options, you won't be able to move forward until the correct answer is selected.

*Click **Next** when you're ready to begin.*

Back

Next

Google

Knowledge check 1

A state consists of a single _____ or string that describes a device's condition, status, setting, position, mode, etc.

Fill in the blank.
Select the best answer from the options listed below.

enum	word
field	subfield

Back

Next

Knowledge check 1

A state consists of a single _____ or string that describes a device's condition, status, setting, position, mode, etc.

Fill in the blank.
Select the best answer from the options listed below.

enum	word
field	subfield

Close... but not quite right! 🤔

Try again

Back

Next

Knowledge check 1

A state consists of a single _____ or string that describes a device's condition, status, setting, position, mode, etc.

Fill in the blank.
Select the best answer from the options listed below.

enum

word

field

subfield

That's right! 🎉

Like a subfield, a state is simply a **word** or string and its definition.

Examples

ON: Powered on.

OFF: Powered off.

LOW: Low speed or output setting.

MEDIUM: Medium speed or output setting.

HIGH: High speed or output setting.

Back

Next

Knowledge check 1

A state consists of a single _____ or string that describes a device's condition, status, setting, position, mode, etc.

Fill in the blank.
Select the best answer from the options listed below.

enum	word
field	subfield

Close... but not quite right! 🤔

Try again

Back

Next

Knowledge check 1

A state consists of a single _____ or string that describes a device's condition, status, setting, position, mode, etc.

Fill in the blank.
Select the best answer from the options listed below.

enum	word
field	**subfield**

Close... but not quite right! 🤔

Try again

Back

Next

Knowledge check 2

A field is considered _____ if it sends states.

Fill in the blank.
Select the best answer from the options listed below.

an enum

a device

multi-state

an entity type

Back

Next

Knowledge check 2

A field is considered _____ if it sends states.

Fill in the blank.
Select the best answer from the options listed below.

an enum	a device
multi-state	an entity type

Close... but not quite right! 🤔

Try again

Back

Next

Knowledge check 2

A field is considered _____ if it sends states.

Fill in the blank.
Select the best answer from the options listed below.

an enum	a device
multi-state	an entity type

Close... but not quite right! 🤔

Try again

Back

Next

Knowledge check 2

A field is considered _____ if it sends states.

Fill in the blank.
Select the best answer from the options listed below.

an enum	a device
multi-state	an entity type

That's right! 🎉

A **multi-state** is a list of allowed states that can describe a device's condition, status, setting, position, mode, etc.

You can tell if a field is multi-state if one of the following is true:

- The point type subfield is `alarm`, `status`, or `mode`.
- The point type subfield is `command`, and it doesn't have a measurement subfield.

Back

Next

Knowledge check 2

A field is considered a/an _____ if it sends states.

Fill in the blank.
Select the best answer from the options listed below.

enum

device

multi-state

entity type
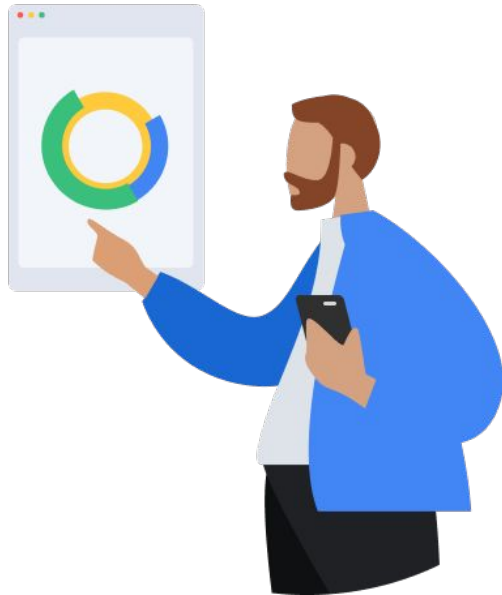
Close... but not quite right! 🤔

Try again

Back

Next

Lesson 5 summary

Let's review what you learned about:

- States
- Multi-states
- State and multi-state constraints

Now you should be able to:

- Describe the concepts of a state and multi-state.
- Identify a multi-state in source code.
- Understand the rules and constraints for multi-states.

Back

Next

You completed Lesson 5!

Now's a great time to take a quick break before starting Lesson 6.

Ready for Lesson 6?

Let's go!

Back

Google

Have questions?

For future reference, keep these contacts and resources easily accessible for technical and procedural questions.

Key contacts

- For DBO questions: Trevor (tsodorff@) or Charbel (charbelk@)
- For UDMI questions: udmi-discuss@

Helpful resources

Bookmark these resources for future reference.

- states.yaml
  Contains all of the available states.
- digitalbuildings / ontology
  Contains the documentation and configuration files for the DBO.
- Ontology Overview
  Provides an overview of the structure and principles of the ontology.
- Model
  Describes the conventions used in the DBO abstract model.