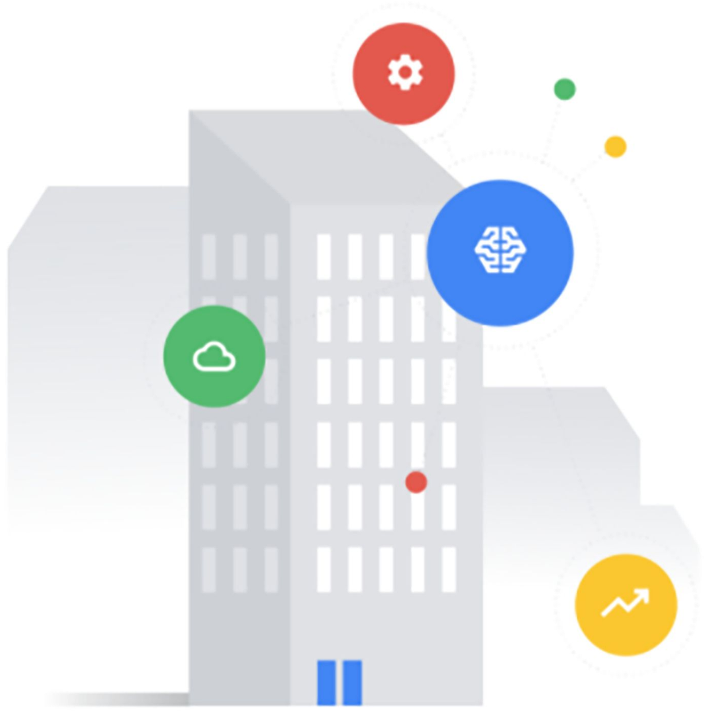
Google

Module 1 | Lesson 9

Digital Buildings Ontology (DBO)



Before you get started

This learning module 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 content.

1 View this content outside of GitHub.

- For the best learning experience, you're encouraged to download a copy so links and other interactive features will be enabled.
- To download a copy of this lesson, click **Download** in the top-right corner of this content block.
- After downloading, open the file in your preferred PDF reader application.

2 Navigate by clicking the buttons and links.

- For the best learning experience, using your keyboard or mouse wheel to navigate is discouraged. However, this is your only option if you're viewing from GitHub.
- If you're viewing this content outside of GitHub:
  - 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.

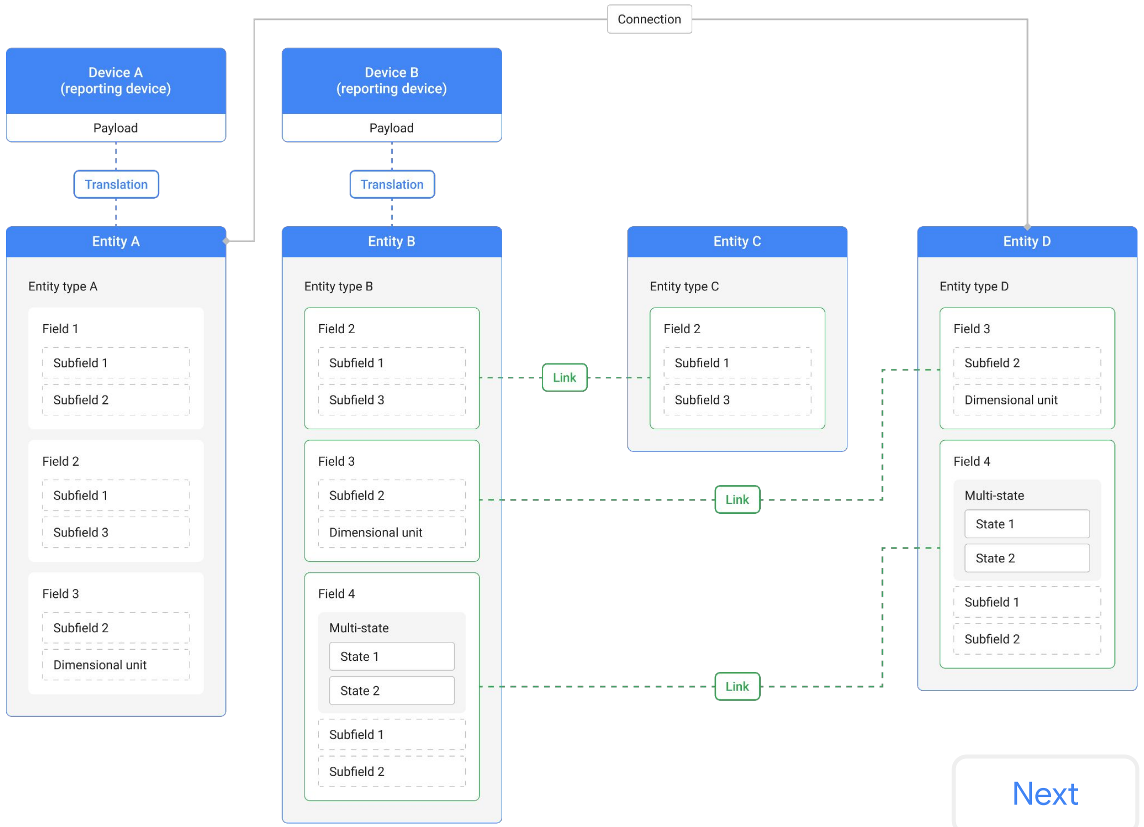
Ready to get started? Let's go!

Google

Conceptual model revisited

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

In this lesson, you'll explore how the abstract and concrete modeling concepts are organized into namespaces.



Back

Next

Lesson 9

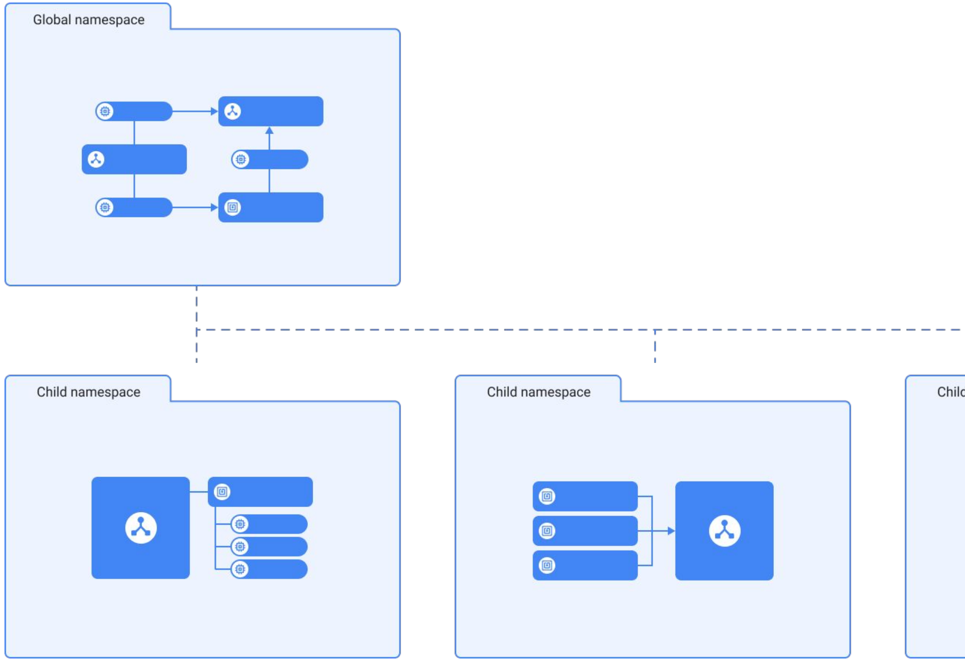
Namespaces

What you'll learn about:

- Global and child namespaces
- Namespace folder structure
- Namespace qualification rules

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

- Describe the concept of a namespace.
- Explain the difference between the global namespace and a child namespace.
- Navigate the namespace folder structure.
- Qualify a concept depending on its namespace.



Back

Next

Google

What’s a namespace?

A namespace is an organizational concept for curating abstract modeling concepts.

You can think of a namespace as a folder. We’ve organized the DBO using namespace folders to make it easier for modelers, like yourself, to find and retrieve the modeling concepts needed for data modeling.

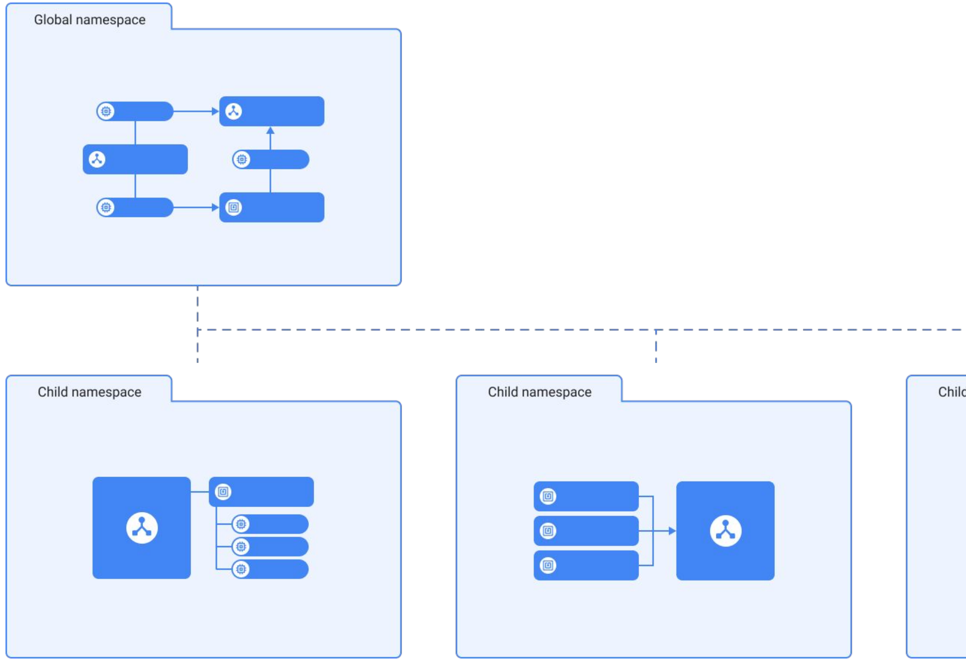
The DBO is structured into two levels of namespaces.

Global namespace

The global namespace contains modeling concepts that are reusable across all application domains without namespace qualification.

Child namespaces

Child namespaces are named after application domains (e.g., HVAC or LIGHTING). They contain locally-defined modeling concepts that are particular to that domain.



Back

Next

Google

Folder structure

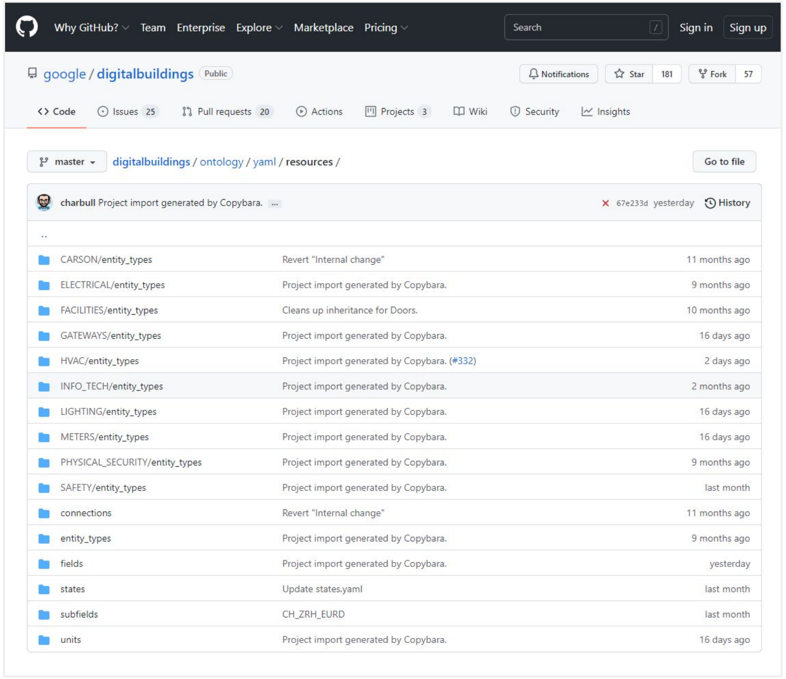
The DBO namespaces are set up in the Digital Buildings Project's GitHub repository.

Click on each level of the folder structure to reveal information.

Top-level folder

Global namespace folders

Child [NAMESPACE] folders



digitalbuildings / ontology / yaml / resources /

charbull Project import generated by Copybara. ... 67e233d yesterday History

..

CARSON/entity_types	Revert "Internal change"	11 months ago
ELECTRICAL/entity_types	Project import generated by Copybara.	9 months ago
FACILITIES/entity_types	Cleans up inheritance for Doors.	10 months ago
GATEWAYS/entity_types	Project import generated by Copybara.	16 days ago
HVAC/entity_types	Project import generated by Copybara. (#332)	2 days ago
INFO_TECH/entity_types	Project import generated by Copybara.	2 months ago
LIGHTING/entity_types	Project import generated by Copybara.	16 days ago
METERS/entity_types	Project import generated by Copybara.	16 days ago
PHYSICAL_SECURITY/entity_types	Project import generated by Copybara.	9 months ago
SAFETY/entity_types	Project import generated by Copybara.	last month
connections	Revert "Internal change"	11 months ago
entity_types	Project import generated by Copybara.	9 months ago
fields	Project import generated by Copybara.	yesterday
states	Update states.yaml	last month
subfields	CH_ZRH_EURD	last month
units	Project import generated by Copybara.	16 days ago

Back

Next

Folder structure

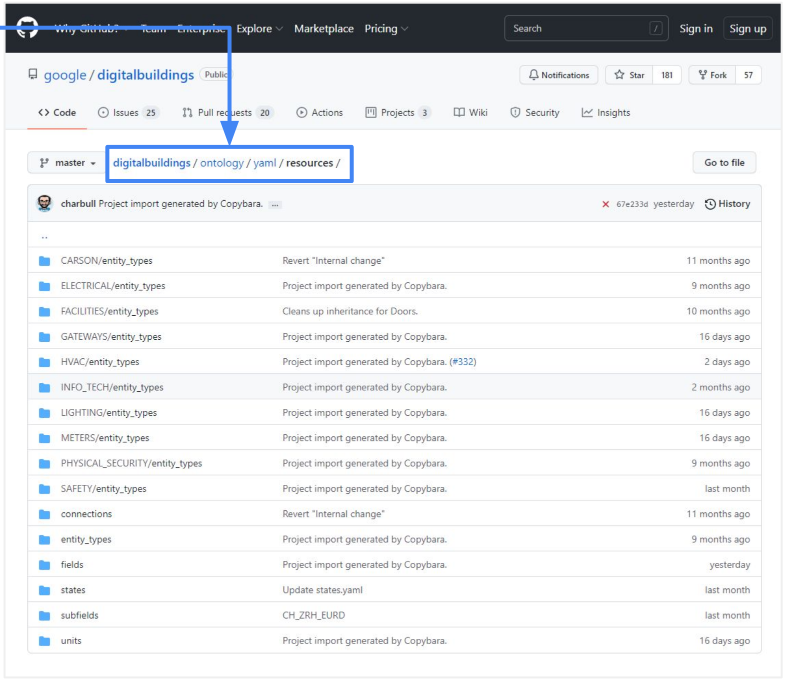
The DBO namespaces are set up in the Digital Buildings Project's GitHub repository.

Click on each level of the folder structure to reveal information.

Top-level folder

Global namespace folders

Child [NAMESPACE] folders



digitalbuildings / ontology / yaml / resources /

charbull Project import generated by Copybara. ... 67e233d yesterday

..		
CARSON/entity_types	Revert "Internal change"	11 months ago
ELECTRICAL/entity_types	Project import generated by Copybara.	9 months ago
FACILITIES/entity_types	Cleans up inheritance for Doors.	10 months ago
GATEWAYS/entity_types	Project import generated by Copybara.	16 days ago
HVAC/entity_types	Project import generated by Copybara. (#332)	2 days ago
INFO_TECH/entity_types	Project import generated by Copybara.	2 months ago
LIGHTING/entity_types	Project import generated by Copybara.	16 days ago
METERS/entity_types	Project import generated by Copybara.	16 days ago
PHYSICAL_SECURITY/entity_types	Project import generated by Copybara.	9 months ago
SAFETY/entity_types	Project import generated by Copybara.	last month
connections	Revert "Internal change"	11 months ago
entity_types	Project import generated by Copybara.	9 months ago
fields	Project import generated by Copybara.	yesterday
states	Update states.yaml	last month
subfields	CH_ZRH_EURD	last month
units	Project import generated by Copybara.	16 days ago

Top-level folder

This contains all concepts of the DBO defined in YAML format.

Back

Next

Google

Folder structure

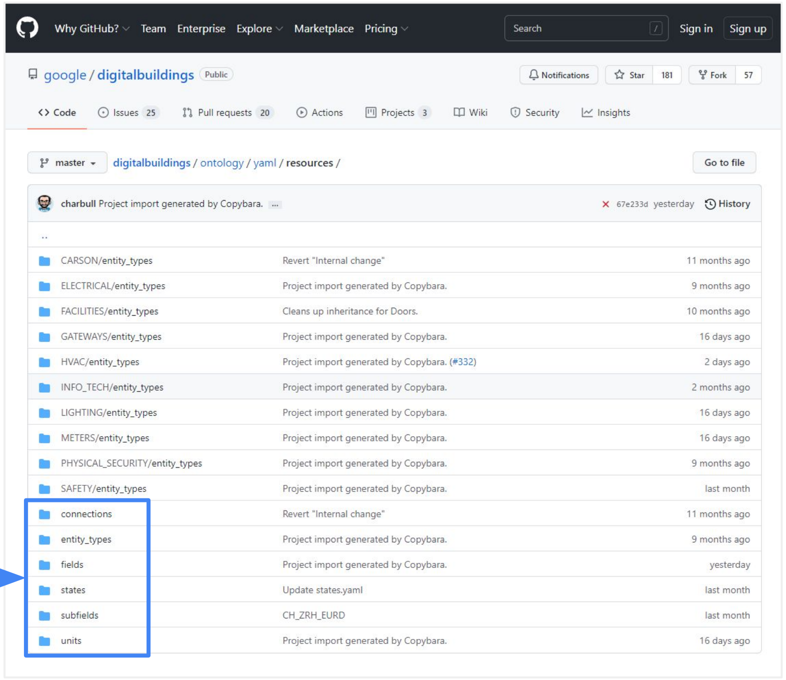
The DBO namespaces are set up in the Digital Buildings Project's GitHub repository.

Click on each level of the folder structure to reveal information.

Top-level folder

Global namespace folders

Child [NAMESPACE] folders



google / digitalbuildings Public

digitalbuildings / ontology / yaml / resources /

charbull Project import generated by Copybara. … 67e233d yesterday

..

CARSON/entity_types	Revert "Internal change"	11 months ago
ELECTRICAL/entity_types	Project import generated by Copybara.	9 months ago
FACILITIES/entity_types	Cleans up inheritance for Doors.	10 months ago
GATEWAYS/entity_types	Project import generated by Copybara.	16 days ago
HVAC/entity_types	Project import generated by Copybara. (#332)	2 days ago
INFO_TECH/entity_types	Project import generated by Copybara.	2 months ago
LIGHTING/entity_types	Project import generated by Copybara.	16 days ago
METERS/entity_types	Project import generated by Copybara.	16 days ago
PHYSICAL_SECURITY/entity_types	Project import generated by Copybara.	9 months ago
SAFETY/entity_types	Project import generated by Copybara.	last month
connections	Revert "Internal change"	11 months ago
entity_types	Project import generated by Copybara.	9 months ago
fields	Project import generated by Copybara.	yesterday
states	Update states.yaml	last month
subfields	CH_ZRH_EURD	last month
units	Project import generated by Copybara.	16 days ago

Global namespace folders

The global namespace includes folders containing curated subfields, units, states, fields, entity types, and connections that are reusable across all namespaces.

These modeling concepts are broadly defined, making them applicable to reuse by any application domain.

Note that we prefer to define subfields, units, states, and fields in the global namespace to encourage reuse.

Back

Next

Google

Folder structure

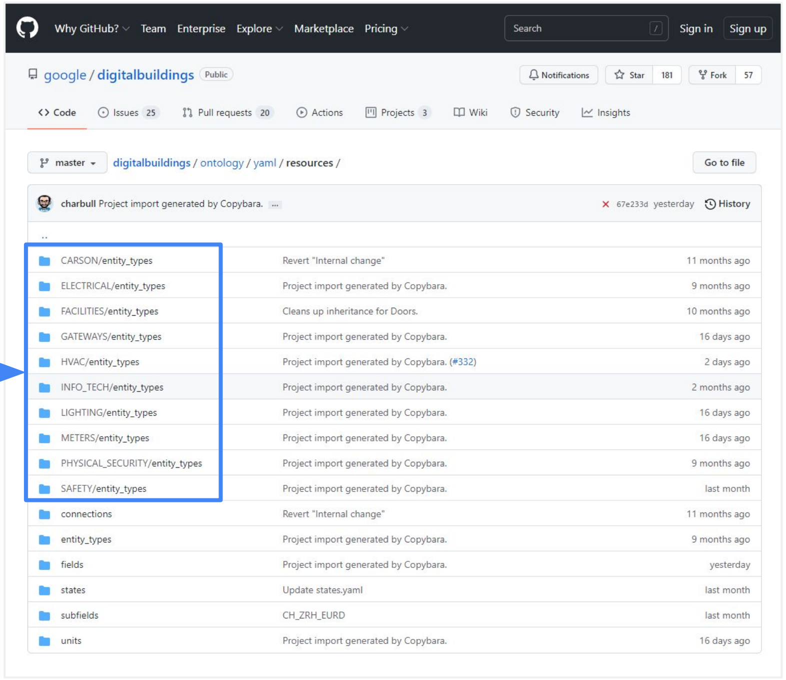
The DBO namespaces are set up in the Digital Buildings Project's GitHub repository.

Click on each level of the folder structure to reveal information.

Top-level folder

Global namespace folders

Child [NAMESPACE] folders



digitalbuildings / ontology / yaml / resources /

charbull Project import generated by Copybara. … 67e233d yesterday

..		
CARSON/entity_types	Revert "Internal change"	11 months ago
ELECTRICAL/entity_types	Project import generated by Copybara.	9 months ago
FACILITIES/entity_types	Cleans up inheritance for Doors.	10 months ago
GATEWAYS/entity_types	Project import generated by Copybara.	16 days ago
HVAC/entity_types	Project import generated by Copybara. (#332)	2 days ago
INFO_TECH/entity_types	Project import generated by Copybara.	2 months ago
LIGHTING/entity_types	Project import generated by Copybara.	16 days ago
METERS/entity_types	Project import generated by Copybara.	16 days ago
PHYSICAL_SECURITY/entity_types	Project import generated by Copybara.	9 months ago
SAFETY/entity_types	Project import generated by Copybara.	last month
connections	Revert "Internal change"	11 months ago
entity_types	Project import generated by Copybara.	9 months ago
fields	Project import generated by Copybara.	yesterday
states	Update states.yaml	last month
subfields	CH_ZRH_EURD	last month
units	Project import generated by Copybara.	16 days ago

Child namespace folders

Child namespaces include folders named after specific application domains like HVAC, ELECTRICAL, and LIGHTING.

Curated canonical, general, and abstract entity types are defined in child namespace folders. These types are specific to the application domain of its child namespace and can be reused with proper qualification.

Note that modeling concepts besides entity types are rarely defined in a child namespace.

Back

Next

Google

Namespace considerations

Consider the following when working with namespaces.

There are only two levels of namespaces.

The top-level of the DBO is the global namespace. Below the global namespace are child namespaces. Hierarchical namespacing within child namespace folders isn't allowed.

Example

There are several different lighting system brands. Creating an additional namespace folder for each brand in the LIGHTING namespace isn't supported by the DBO.

Instead, a type of lighting system should be defined in the LIGHTING child namespace regardless of their brand.

A concept can be defined in different namespaces.

Sometimes, concepts may have an identical name but unique definitions for different application domains. The abstract entity type `SS` is an example of a concept that has been uniquely defined in several different namespaces.

In the global namespace, the abstract type `SS` defines the start/stop run command and status for an arbitrary device. It is broadly defined for easy reuse across namespaces.

There are other `SS` abstract types that are uniquely defined in child namespaces like `HVAC/SS` and `LIGHTING/SS`. Each one is a unique concept with an extended definition of `SS` that's relevant to its specific application domain.

- `HVAC/SS` inherits from the global `SS` and adds fields like `power_sensor` and `current_sensor`, which are relevant to devices in the HVAC namespace.
- `LIGHTING/SS` also inherits from the global `SS` and could contain a field like `illuminance_sensor`, which may be useful for the control of a light but would be nonsensical for a pump.

Back

Next

Google

Namespace qualification

Sometimes, a concept needs to be referenced that isn't defined in the same namespace as the device being modeled.

When referencing concepts in different namespaces, you need to qualify the namespace where the concept is defined. This requires **namespace qualification**, which is similar to identifying the file path of a file's location in a directory.

Qualification rules

1. Globally-defined fields are exempt from qualification rules.
2. To qualify an adjacent child namespace, the namespace must be specified before stating the concept.

   `[ADJACENT_NAMESPACE]/[type]`
3. To qualify the global namespace within a child namespace, omit anything before the delimiter.

   `/[type]`
4. No qualification is needed to reference concepts in the same namespace.

   `[type]`

Note: If the referenced concept is unqualified and isn't defined in the same namespace, then the unqualified concept is assumed to be from the global namespace.

Example 1

The entity type **DFSMC** is defined in the HVAC namespace.

It has two required field associations: **`discharge_fan_run_command`** and **`discharge_fan_speed_mode`**. Both fields are defined in the global namespace and don't require namespace qualification to be referenced.

```
DFSMC:
  id: "2774887660236308480"
  description: "Discharge fan multi-speed control."
  is_abstract: true
  uses:
  - discharge_fan_run_command
  - discharge_fan_speed_mode
```

Back

Note: Qualification rules apply after a field undergoes the process of namespace elevation. As an MSI, you should be aware of the effects of namespace elevation, but you'll rarely ever find yourself elevating field definitions. You should stick with convention and adhere to these qualification rules.

Next

Google

Namespace qualification (continued)

Sometimes, a concept needs to be referenced that isn't defined in the same namespace as the device being modeled.

When referencing concepts in different namespaces, you need to qualify the namespace where the concept is defined. This requires **namespace qualification**, which is similar to identifying the file path of a file's location in a directory.

Qualification rules

1. Globally-defined fields are exempt from qualification rules.
2. To qualify an adjacent child namespace, the namespace must be specified before stating the concept.

   `[ADJACENT_NAMESPACE]/[type]`

3. To qualify the global namespace within a child namespace, omit anything before the delimiter.

   `/[type]`

4. No qualification is needed to reference concepts in the same namespace.

   `[type]`

Note: If the referenced concept is unqualified and isn't defined in the same namespace, then the unqualified concept is assumed to be from the global namespace.

Example 2

The entity type DOOR_EMER is defined in the PHYSICAL_SECURITY namespace.

It has a parent type, DOOR, which is defined in the FACILITIES namespace. Using the namespace qualifier **`FACILITIES/`**, the entity type **`DOOR_EMER`** references the namespace where the parent type is defined. Without qualification, this entity type would be invalid.

```
DOOR_EMER:
  id: "2344230945869004800"
  description: "A door with position and emergency
release monitoring."
  is_canonical: true
  implements:
  - FACILITIES/DOOR
  uses:
  - position_status
  - emergency_release_status
```

Back

Note: Qualification rules apply after a field undergoes the process of namespace elevation. As an MSI, you should be aware of the effects of namespace elevation, but you'll rarely ever find yourself elevating field definitions. You should stick with convention and adhere to these qualification rules.

Next

Google

Namespace qualification (continued)

Sometimes, a concept needs to be referenced that isn't defined in the same namespace as the device being modeled.

When referencing concepts in different namespaces, you need to qualify the namespace where the concept is defined. This requires **namespace qualification**, which is similar to identifying the file path of a file's location in a directory.

Qualification rules

1. Globally-defined fields are exempt from qualification rules.
2. To qualify an adjacent child namespace, the namespace must be specified before stating the concept.

   `[ADJACENT_NAMESPACE]/[type]`

3. To qualify the global namespace within a child namespace, omit anything before the delimiter.

   `/[type]`

4. No qualification is needed to reference concepts in the same namespace.

   `[type]`

Note: If the referenced concept is unqualified and isn't defined in the same namespace, then the unqualified concept is assumed to be from the global namespace.

Example 3

The entity type `SS` is defined in the LIGHTING namespace. We'll refer to it as `LIGHTING/SS`.

It has a parent type, `SS`, which is defined in the global namespace. These are two unique concepts.

Using `/` to qualify the global namespace, the entity type `LIGHTING/SS` references the namespace where the global `SS` is defined. Without qualification, this entity type would be invalid.

```
SS:
  id: "7172089746410176512"
  description: "Basic combination of run command and
status (start/stop); indicates the light is active or
inactive."
  is_abstract: true
  implements:
  - /SS
```

Back

Note: Qualification rules apply after a field undergoes the process of namespace elevation. As an MSI, you should be aware of the effects of namespace elevation, but you'll rarely ever find yourself elevating field definitions. You should stick with convention and adhere to these qualification rules.

Next

Google

Namespace qualification (continued)

Sometimes, a concept needs to be referenced that isn't defined in the same namespace as the device being modeled.

When referencing concepts in different namespaces, you need to qualify the namespace where the concept is defined. This requires **namespace qualification**, which is similar to identifying the file path of a file's location in a directory.

Qualification rules

1. Globally-defined fields are exempt from qualification rules.
2. To qualify an adjacent child namespace, the namespace must be specified before stating the concept.

   `[ADJACENT_NAMESPACE]/[type]`

3. To qualify the global namespace within a child namespace, omit anything before the delimiter.

   `/[type]`

4. No qualification is needed to reference concepts within the same namespace.

   `[type]`

   Note: If the referenced concept is unqualified and isn't defined in the same namespace, then the unqualified concept is assumed to be from the global namespace.

Example 4

The entity type **`AHU_DFSS_DSP_DXZC`** is defined in the HVAC namespace.

It has several parent types, which are all defined in the HVAC namespace, too. The entity type **`AHU_DFSS_DSP_DXZC`** requires no qualification to reference these parent types.

```
AHU_DFSS_DSP_DXZC:
  id: "12007351558741164032"
  description: "Single zone AHU."
  is_canonical: true
  implements:
  - AHU
  - DFSS
  - DSP
  - DXZC
```

Back

Note: Qualification rules apply after a field undergoes the process of namespace elevation. As an MSI, you should be aware of the effects of namespace elevation, but you'll rarely ever find yourself elevating field definitions. You should stick with convention and adhere to these qualification rules.

Next

Lesson 9

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.

Back

Google

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

Next

Knowledge check 1

A child namespace contains curated abstract modeling concepts with specific definitions that are particular to an application domain.

Which namespace contains broadly defined modeling concepts?
Select the best answer from the options listed below.

Global namespace

ELECTRICAL namespace

HVAC namespace

INFO_TECH namespace

Back

Next

Google

Knowledge check 1

A child namespace contains curated abstract modeling concepts with specific definitions that are particular to an application domain.

Which namespace contains broadly defined modeling concepts?
Select the best answer from the options listed below.

[✓ Global namespace]
ELECTRICAL namespace
HVAC namespace
INFO_TECH namespace

That's right! 🎉

The **global namespace** contains broadly defined modeling concepts that are applicable for reuse by any application domain.

We prefer to define subfields, units, states, fields, and connection types in the global namespace to encourage reuse. These modeling concepts are rarely defined in child namespaces.

Curated entity types can also be defined in the global namespace. However, you'll find most of them are contained within a relevant child namespace and have specific definitions that are particular to an application domain.

Back

Next

Google

Knowledge check 1

A child namespace contains curated abstract modeling concepts with specific definitions that are particular to an application domain.

Which namespace contains broadly defined modeling concepts?
Select the best answer from the options listed below.

Global namespace [✓ ELECTRICAL namespace]
HVAC namespace
INFO_TECH namespace

Hmm, that's not right! 🤔

The **ELECTRICAL namespace** is a child namespace that contains curated entity types that are specific to the electrical application domain including uninterruptible power supplies.

Try again

Back

Next

Google

Knowledge check 1

A child namespace contains curated abstract modeling concepts with specific definitions that are particular to an application domain.

Which namespace contains broadly defined modeling concepts?
Select the best answer from the options listed below.

Global namespace
ELECTRICAL namespace
[✓ HVAC namespace]
INFO_TECH namespace

Hmm, that's not right! 🤔

The **HVAC namespace** is a child namespace that contains curated entity types that are specific to the HVAC application domain including air handling units.

Try again

Back

Next

Google

Knowledge check 1

A child namespace contains curated abstract modeling concepts with specific definitions that are particular to an application domain.

Which namespace contains broadly defined modeling concepts?
Select the best answer from the options listed below.

Global namespace
ELECTRICAL namespace
HVAC namespace
[✓ INFO_TECH namespace]

Hmm, that's not right! 🤔

The **INFO_TECH namespace** is a child namespace that contains curated entity types that are specific to the information technology application domain including printers.

Try again

Back

Next

Google

Knowledge check 2

Let’s say you have two different types of lighting system (DALI and ENLIGHTED).

You want to add new namespace directories for them within the LIGHTING namespace (e.g., `LIGHTING/DALI/` and `LIGHTING/ENLIGHTED/`).

Is this allowed?

Select the best answer from the options listed below.

Yes

No

Back

Next

Google

Knowledge check 2

Let’s say you have two different types of lighting system (DALI and ENLIGHTED).

You want to add new namespace directories for them within the LIGHTING namespace (e.g., `LIGHTING/DALI/` and `LIGHTING/ENLIGHTED/`).

Is this allowed?
Select the best answer from the options listed below.

[✓ Yes]

No

Hmm, that’s not right! 🤔

There are only two levels of namespaces in the DBO:
- Global namespace
- Child namespace

Hierarchical namespacing within child namespace folders like LIGHTING isn’t allowed.

Back

Next

Knowledge check 2

Let's say you have two different types of lighting system (DALI and ENLIGHTED).

You want to add new namespace directories for them within the LIGHTING namespace (e.g., `LIGHTING/DALI/` and `LIGHTING/ENLIGHTED/`).

Is this allowed?
Select the best answer from the options listed below.

Yes [✓ No]

That's right! 🎉

There are only two levels of namespaces in the DBO:

- Global namespace
- Child namespace

Hierarchical namespacing within child namespace folders like LIGHTING isn't allowed.

Back   Next

Google

Knowledge check 3

Let's say you're creating a new canonical type within the HVAC namespace. It implements the abstract type `OPERATIONAL`, which is defined in the HVAC namespace.

What is the correct way to qualify `OPERATIONAL`?
Select the best answer from the options listed below.

```
SPSS_...:
  description: "Spray pump start stop monitoring."
  is_canonical: true
  uses:
  - spray_pump_run_command
  - spray_pump_run_status
  implements:
  - ???
```

`HVAC/OPERATIONAL`

`ANALYSIS/OPERATIONAL`

`/OPERATIONAL`

`OPERATIONAL`

Back

Next

Google

Knowledge check 3

Let’s say you’re creating a new canonical type within the HVAC namespace. It implements the abstract type `OPERATIONAL`, which is defined in the HVAC namespace.

What is the correct way to qualify `OPERATIONAL`?
Select the best answer from the options listed below.

[✓ `HVAC/OPERATIONAL`] `ANALYSIS/OPERATIONAL`
`/OPERATIONAL` `OPERATIONAL`

Hmm, that’s not right! 🤔

You don’t need to qualify the HVAC namespace, since you’re working within the same namespace.

```
SPSS_...:
  description: "Spray pump start stop monitoring."
  is_canonical: true
  uses:
  - spray_pump_run_command
  - spray_pump_run_status
  implements:
  - HVAC/OPERATIONAL
```

Try again

Back Next

Knowledge check 3

Let's say you're creating a new canonical type within the HVAC namespace. It implements the abstract type `OPERATIONAL`, which is defined in the HVAC namespace.

What is the correct way to qualify `OPERATIONAL`?

Select the best answer from the options listed below.

HVAC/OPERATIONAL

[✓ ANALYSIS/OPERATIONAL]

/OPERATIONAL

OPERATIONAL

Hmm, that's not right! 🤔

You'll find **`OPERATIONAL`** in the ANALYSIS.yaml. However, you don't need to qualify the file name where it's contained.

```
SPSS_...:
  description: "Spray pump start stop monitoring."
  is_canonical: true
  uses:
  - spray_pump_run_command
  - spray_pump_run_status
  implements:
  - ANALYSIS/OPERATIONAL
```

Try again

Back

Next

Google

Knowledge check 3

Let’s say you’re creating a new canonical type within the HVAC namespace. It implements the abstract type `OPERATIONAL`, which is defined in the HVAC namespace.

What is the correct way to qualify `OPERATIONAL`?
Select the best answer from the options listed below.

- HVAC/OPERATIONAL
- ANALYSIS/OPERATIONAL
- [✓ /OPERATIONAL]
- OPERATIONAL

Hmm, that’s not right! 🤔

Since you’re within the HVAC namespace, putting the delimiter before **`OPERATIONAL`** indicates you’re qualifying the global namespace.

```
SPSS_...:
  description: "Spray pump start stop monitoring."
  is_canonical: true
  uses:
  - spray_pump_run_command
  - spray_pump_run_status
  implements:
  - /OPERATIONAL
```

Try again

Back

Next

Knowledge check 3

Let's say you're creating a new canonical type within the HVAC namespace. It implements the abstract type **OPERATIONAL**, which is defined in the HVAC namespace.

What is the correct way to qualify OPERATIONAL?

Select the best answer from the options listed below.

HVAC/OPERATIONAL

ANALYSIS/OPERATIONAL

/OPERATIONAL

[✓ OPERATIONAL]

That's right! 🎉

No qualification is needed to reference concepts within the same namespace. Here's a sample entity type defined in the HVAC namespace that properly qualifies **OPERATIONAL**.

```
SPSS_...:
  description: "Spray pump start stop monitoring."
  is_canonical: true
  uses:
  - spray_pump_run_command
  - spray_pump_run_status
  implements:
  - OPERATIONAL
```

Back

Next

Google

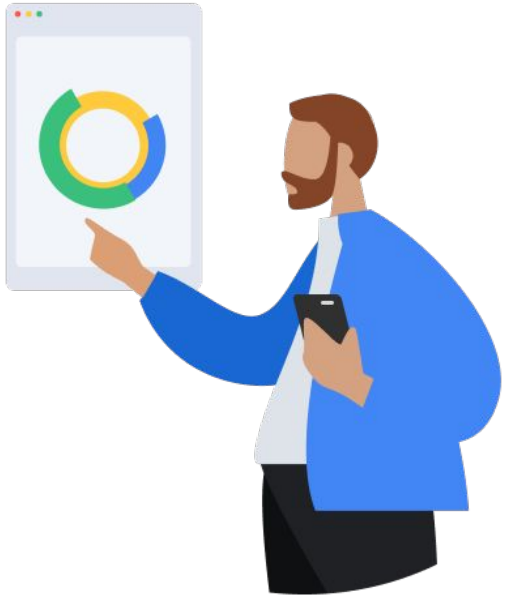
Lesson 9 summary

Let’s review what you learned about:

- Global and child namespaces
- Namespace folder structure
- Namespace qualification rules

Now you should be able to:

- Describe the concept of a namespace.
- Explain the difference between the global namespace and a child namespace.
- Navigate the namespace folder structure.
- Qualify a concept depending on its namespace.



You completed Lesson 9!

This also completes Module 1:
Digital Buildings Ontology (DBO).

Back

Helpful resources

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

- digitalbuildings / ontology / yaml / resources
  Contains the global and child namespaces of the DBO.
- digitalbuildings / ontology / yaml / ontology_config.md
  Explains how to write YAML configuration files using what's provided in the global and child namespaces.
- Digital Buildings Project GitHub
  Contains source code, tooling, and documentation for the DBO.