

BAS TAG MODELLING STANDARDS

Rev 1.4

Contents

Document Control.....	3
Source of truth	4
Space Management System.....	4
Identification.....	4
Relationships.....	4
Tags	4
Asset Codes	4
Identification.....	4
Relationships.....	4
BAS Relationship Modelling.....	5
Definition of terms.....	5
Geographical	5
Hierarchical	5
Asset.....	5
Point.....	5
Cyclic Relationships.....	5
Containment Relationships.....	6
Serves/Flow Relationships	7
Space and Zone Modelling Concepts.....	8
Brick Schema Minimum Required Relationships	11
Relationship Domains	11
Containment Relationships.....	14
Serves/Flow Relationships	15
Haystack Minimum Required Relationships	17
Relationship Domains	17
Minimum Required Relationships.....	20
Serves/Flow Relationships	21
BAS Tag Modelling	23
Brick Tagging Minimum Requirements.....	23
URI.....	23
Description	23
Equipment.....	24

Point.....	25
Tag Groups	26
Haystack Tagging Minimum Requirements	27
Unique Id.....	27
URI.....	27
Description (dis Tag)	27
Equipment.....	28
Point.....	29
Tag Groups	30

Document Control

Date Modified	Revision	Tag Schema Versions	Description	Author
9/06/21	1.0		Initial Document	JJ
17/06/21	1.1		Update haystack site and campus modelling, Defined space and zone modelling rules	JJ
11/07/22	1.2		Added additional content for both brick and haystack	JC
18/07/22	1.3		Made some minor changes to the minimum tagging requirements	JC
28/07/22	1.4	Brick v1.2.1 Haystack v3.9.12	Added Brick and Haystack version numbers that this document is based on.	JC

Source of truth

Tagging suppliers must all refer to the same source of truth database when tagging relationships between assets, across different systems. This ensures all entities maintain the same identity and cross system integration is seamless.

Space Management System

For the purposes of this section, it is assumed that the BAS tagging models are running alongside an existing spatial database system.

Identification

All locations can be referenced directly by their Space Management Database id.

E.g. urn:spaceManagement/campus1 – Campus 1

E.g. urn:spaceManagement/campus1/bld10 – Campus 1, Building 10

E.g. urn:spaceManagement/campus1/bld10/lvl3 – Campus 1, Building 10, Level 3

E.g. urn:spaceManagement/campus1/bld10/lvl3/rm12 – Campus 1, Building 10, Level 3, Room 12

Note: The naming pattern pertaining to this example is not indicative of all location-based asset identification standards.

Relationships

Relationships between Space Management entities should not be redefined within controller tag databases.

Tags

Additional tags on Space Management entities are not permitted to be made within controller tag databases as the Space Management database should be the source of truth for space assets.

Asset Codes

Identification

All equipment asset Ids must come from an asset ID source of truth. This is to ensure that all systems referring to equipment remain consistent across multiple databases.

E.g. Space Management, Building Automation Systems, Work Order Systems, etc.

Relationships

Creating relationships between asset coded entities are permitted.

BAS Relationship Modelling

Definition of terms

When creating BAS models, each entity (thing) can be placed into one of the following fundamental categories:

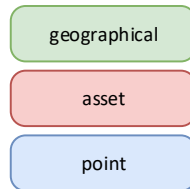


Figure 1 Fundamental BAS Tagging Entity Types

Geographical

Geographical entities represent either the abstract collection of other entities (including spaces) or physical volumes as contained by physical walls.

For example:

- Sites (or Campuses) are a collection of buildings
- Buildings are a collection of floors
- Floors are a collection of rooms and /or zones

Hierarchical

Hierarchical entities represent the grouping of one or more related entities which form part of a larger system or network.

Asset

Assets represent physical equipment in the real world.

Point

Point entities represent software or hardware values within a BAS network.

Cyclic Relationships

Relationship cycle occurs when an entity can traverse 1 or more entity relationships back to itself.

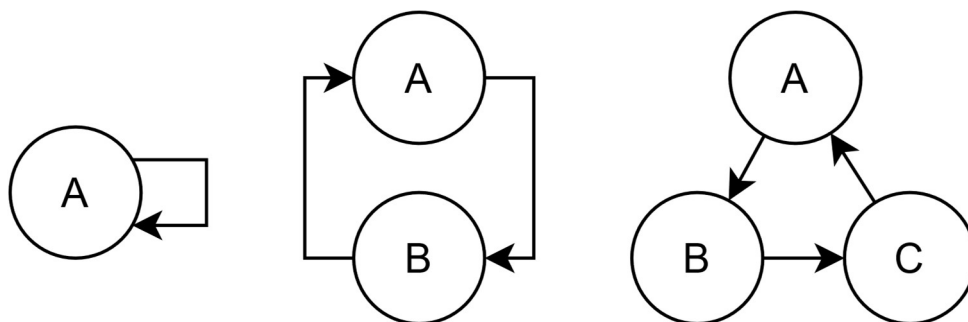


Figure 2 Examples of Cyclic relationship graphs

In most cases, Brick and Haystack have 2 types of relationships, “containment relationships and “serves” relationships.

Containment Relationships

Containment relationships express that one entity hierarchy is part of another entity

The following relationship predicates are containment Relationships

- Brick
 - hasLocation (<https://brickschema.org/ontology/1.2/relationships/hasLocation>)
 - isPartOf (<https://brickschema.org/ontology/1.2/relationships/isPartOf>)
 - isPointOf (<https://brickschema.org/ontology/1.2/relationships/isPointOf>)
- Haystack
 - portfolioRef (proposed only)
 - siteRef (<https://project-haystack.org/doc/lib-phIoT/siteRef>)
 - spaceRef (<https://project-haystack.org/doc/lib-phIoT/spaceRef>)
 - equipRef (<https://project-haystack.org/doc/lib-phIoT/equipRef>)

A containment relationship cycle occurs when any combination of the above containment relationships can traverse back to the same entity.

Cyclic containment relationships **are not** permitted.

Serves/Flow Relationships

Serves relationships indicate the dependence of one entity to another.

The following relationship predicates are Serves Relationships. These are also known as Flow Relationships in Haystack.

- Brick v1.2.1
 - Feeds (<https://brickschema.org/ontology/1.2/relationships/feeds>)
 - Controls (<https://brickschema.org/ontology/1.2/relationships/controls>)
- Haystack v3.9.12
 - airRef(<https://project-haystack.org/doc/lib-phIoT/airRef>)
 - blowdownWaterRef (<https://project-haystack.org/doc/lib-phIoT/blowdownWaterRef>)
 - chilledWaterRef (<https://project-haystack.org/doc/lib-phIoT/chilledWaterRef>)
 - condensateRef (<https://project-haystack.org/doc/lib-phIoT/condensateRef>)
 - condenserWaterRef (<https://project-haystack.org/doc/lib-phIoT/condenserWaterRef>)
 - deviceRef (<https://project-haystack.org/doc/lib-phIoT/deviceRef>)
 - domesticWaterRef (<https://project-haystack.org/doc/lib-phIoT/domesticWaterRef>)
 - elecRef (<https://project-haystack.org/doc/lib-phIoT/elecRef>)
 - fuelOilRef (<https://project-haystack.org/doc/lib-phIoT/fuelOilRef>)
 - gasolineRef (<https://project-haystack.org/doc/lib-phIoT/gasolineRef>)
 - hotWaterRef (<https://project-haystack.org/doc/lib-phIoT/hotWaterRef>)
 - makeupWaterRef (<https://project-haystack.org/doc/lib-phIoT/makeupWaterRef>)
 - naturalGasRef (<https://project-haystack.org/doc/lib-phIoT/naturalGasRef>)
 - networkRef (<https://project-haystack.org/doc/lib-phIoT/networkRef>)
 - refrigRef (<https://project-haystack.org/doc/lib-phIoT/refrigRef>)
 - steamRef (<https://project-haystack.org/doc/lib-phIoT/steamRef>)
 - submeterOf (<https://project-haystack.org/doc/lib-phIoT/submeterOf>)
 - weatherStationRef (<https://project-haystack.org/doc/lib-phIoT/weatherStationRef>)

Cyclic Serves relationships **are** permitted.

Space and Zone Modelling Concepts

Space management systems often only define architectural spaces and zones and may not necessarily define building services zones such as HVAC Zone, Fire Zone, Lighting Zone, etc.

When defining zone and space containment relationships, often the model falls into one of two configurations:

- One space containing one or more zone(s)
- Two or more spaces sharing one or more zone(s)

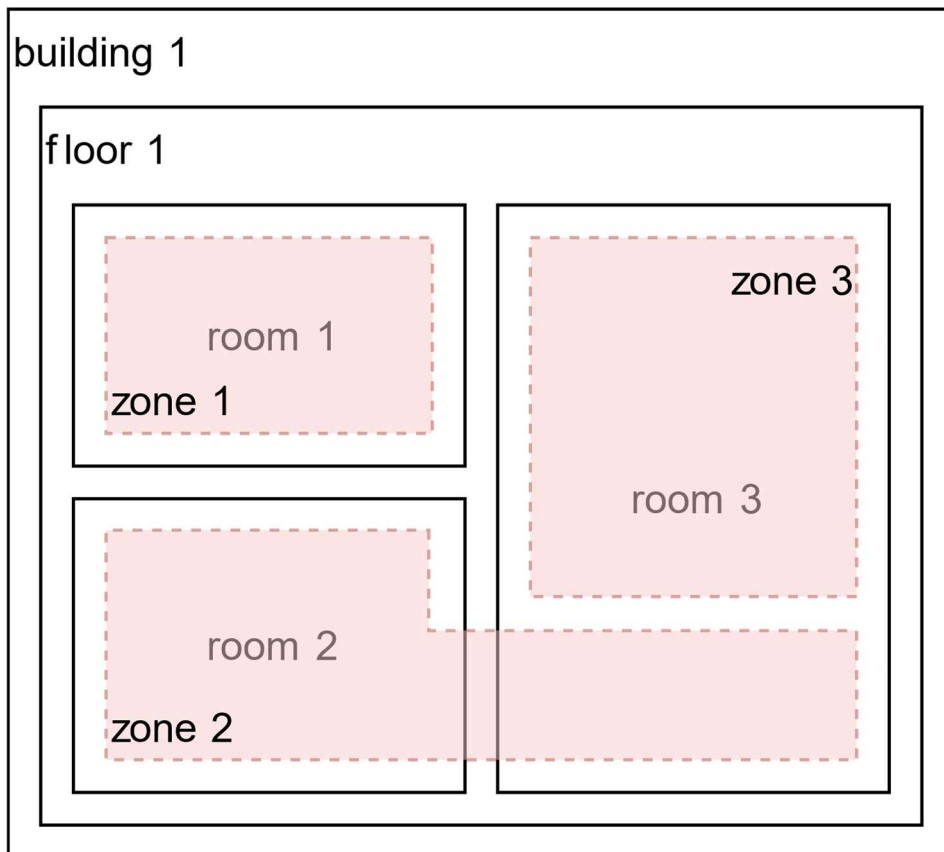


Figure 3 example Zone and space configuration

In the example shown in figure 3, zone 2 is shared by rooms 2 and 3.

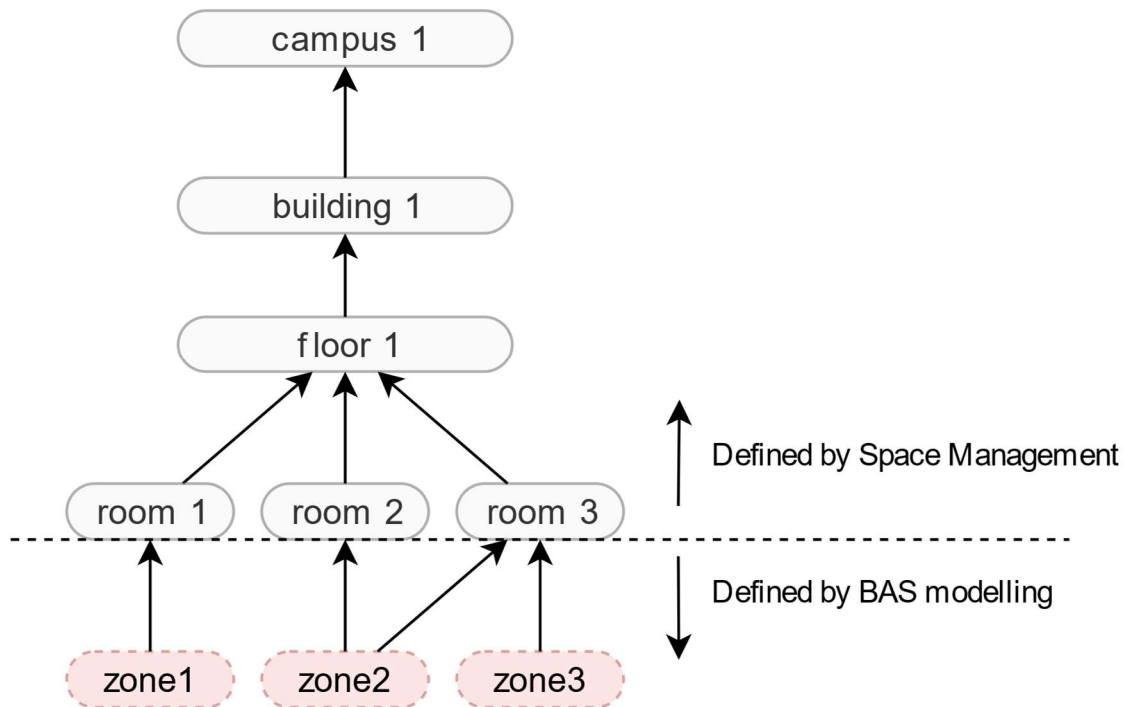


Figure 4 Space nodes are always the roots

If space nodes are always the parents for zones, the space management source of truth is respected and redefinitions of space relationships between the spaces are preserved.

If topological sorting is priorities first, as shown in figure 5 e.g., zone 2 acting as the “parent” containing node for its two child nodes, it may cause invalid redefinitions of space relationships as originally defined in the source of truth.

Therefore, in order to respect the source of truth as managed by space management models, **zones should always be children under spaces**, if the zones are defined from within the BAS model.

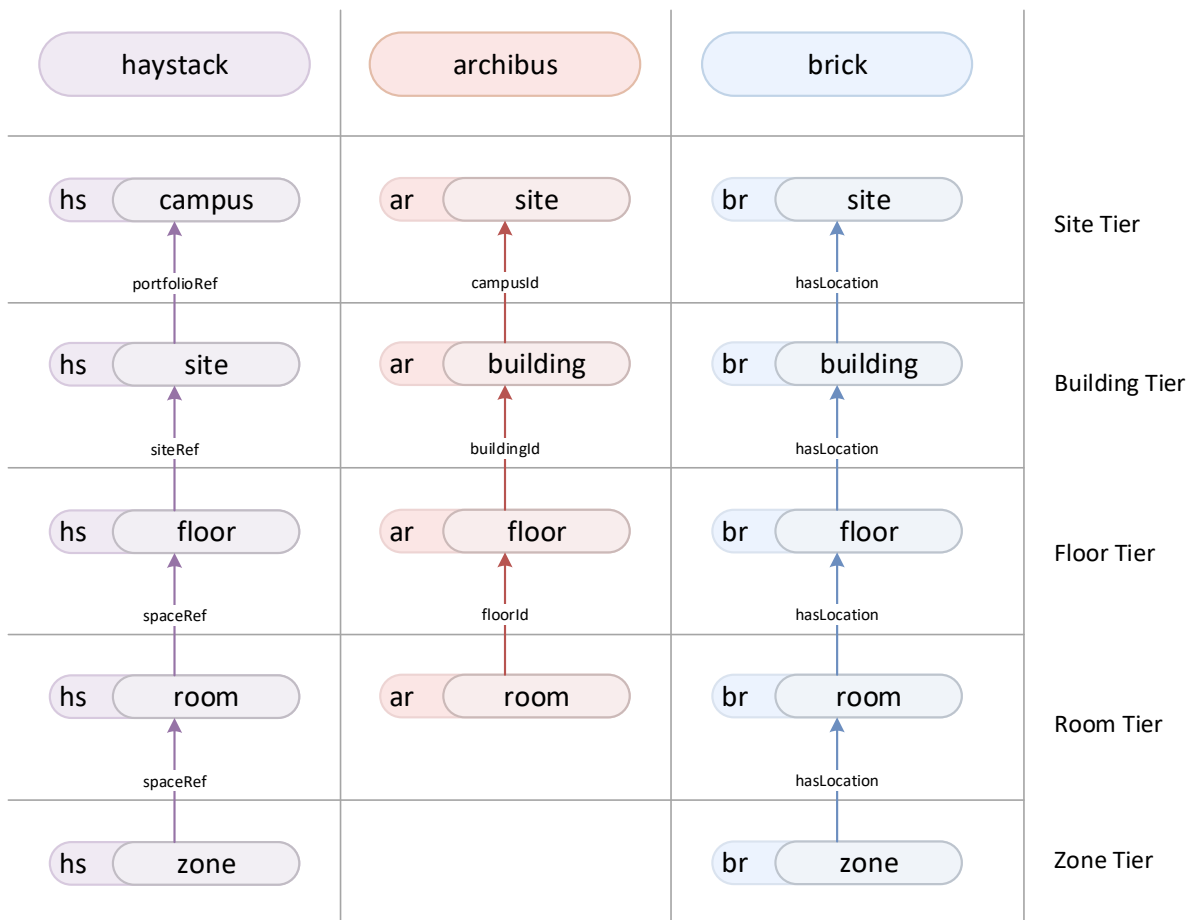


Figure 5 structure of zones and spaces within containment graphs

Brick Schema Minimum Required Relationships

Relationship Domains

To ensure consistency of relationships between brick models, the following constraints on relationship domains are proposed.

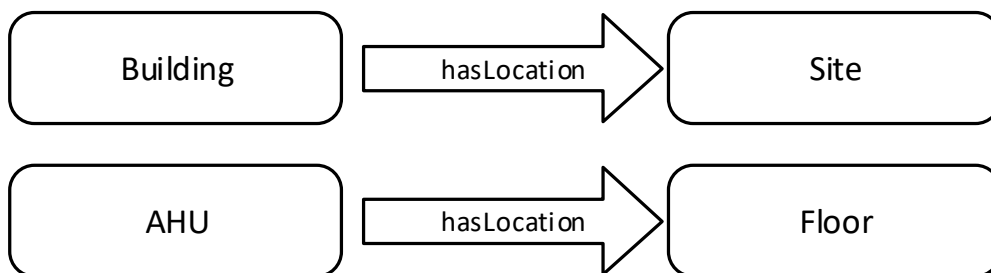
hasLocation

<https://brickschema.org/ontology/1.2/relationships/hasLocation>

Indicates an entity belonging to a space

From Tag	Relationship	To Tag
<ul style="list-style-type: none">• Building• Floor• Room• Zone• Equipment	hasLocation	<ul style="list-style-type: none">• Site• Building• Floor• Room• Zone

Example Use Cases:



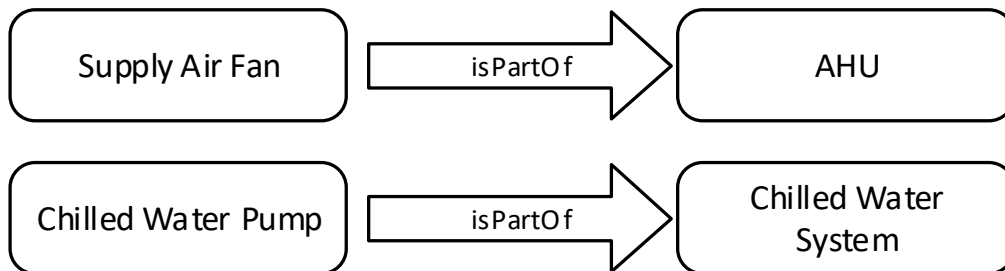
isPartOf

<https://brickschema.org/ontology/1.2/relationships/isPartOf>

Indicates Containment of Sub-Equipment to an Equipment entity.

From Tag	Relationship	To Tag
<ul style="list-style-type: none">• Equipment	isPartOf	Equipment

Example Use cases



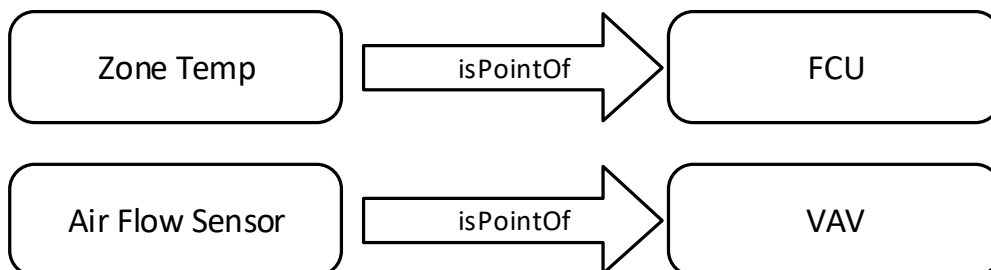
isPointOf

<https://brickschema.org/ontology/1.2/relationships/isPointOf>

Indicates a point is located within a particular Equipment entity.

From Tag	Relationship	To Tag
Point	isPointOf	<ul style="list-style-type: none">• Building• Floor• Equipment

Example Use cases



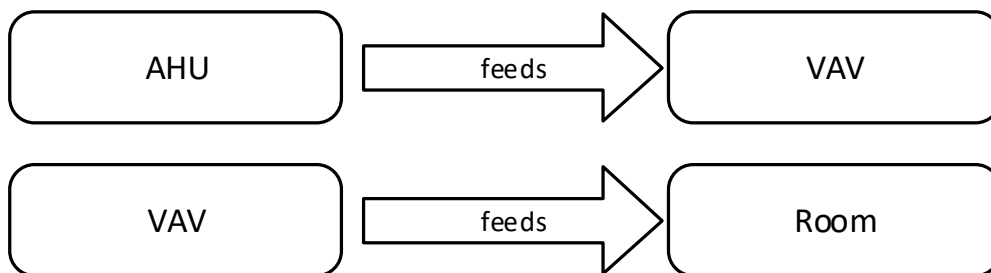
feeds

<https://brickschema.org/ontology/1.2/relationships/feeds>

Indicates the flow of air, water, etc. from the source Equipment Entities to the target Entities which could be other equipment or spaces.

From Tag	Relationship	To Tag
Equipment	feeds	<ul style="list-style-type: none">• Equipment• Floor• Room• Zone

Example Use cases



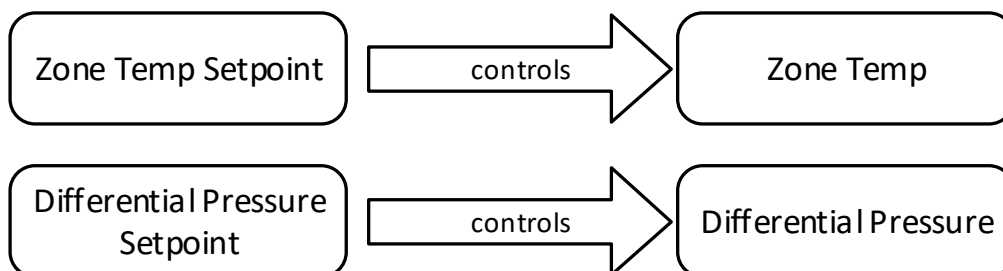
controls

<https://brickschema.org/ontology/1.2/relationships/controls>

Indicates the control of a source software or hardware point to a target software or hardware point.

From Tag	Relationship	To Tag
Point	controls	Point

Example Use cases



Containment Relationships

Brick containment relationship models must follow the following rules:

- All buildings must be located in within a site instance. Linked via a *hasLocation* reference.
- All floors must be located in a building instance. Linked via a *hasLocation* reference.
- Each building must have at least 1 floor.
- Spaces such as rooms and zones within the building can only be associated to their appropriate floors. Linked via a *hasLocation* reference.
- All equipment must be associated with a floor or a space/location within a floor.
- All points must be associated with an equipment and /or space/location.

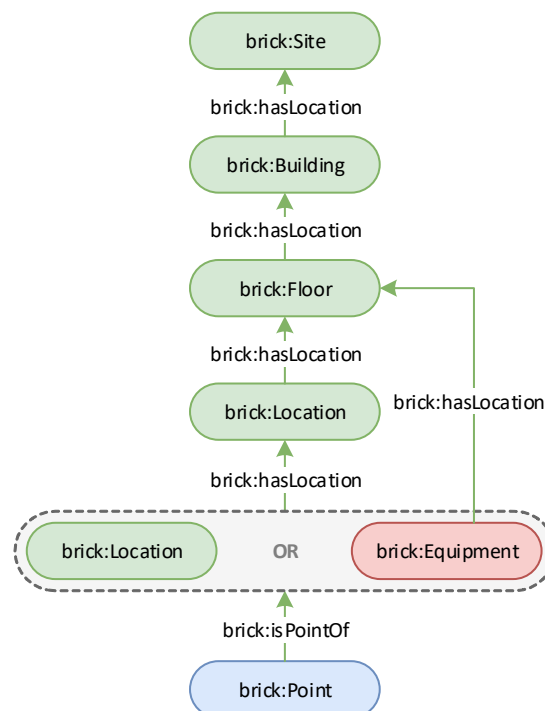


Figure 6 Brick Minimum Containment Relationships

Serves/Flow Relationships

Brick Serves Relationship models must follow the following rules:

- Equipment (Assets) must be associated with the floor or space of where the equipment is located. Linked via a *hasLocation* reference.
- Equipment must be associated with the floor or space that it feeds. Linked via a *feeds* reference.
- Equipment must be associated with Equipment that it feeds. Linked via a *feeds* reference.
- Sub-Equipment must be associated with their parent Equipment. Linked via an *isPartOf* reference.
- Points must be associated with their parent Equipment. Linked via an *isPointOf* reference.

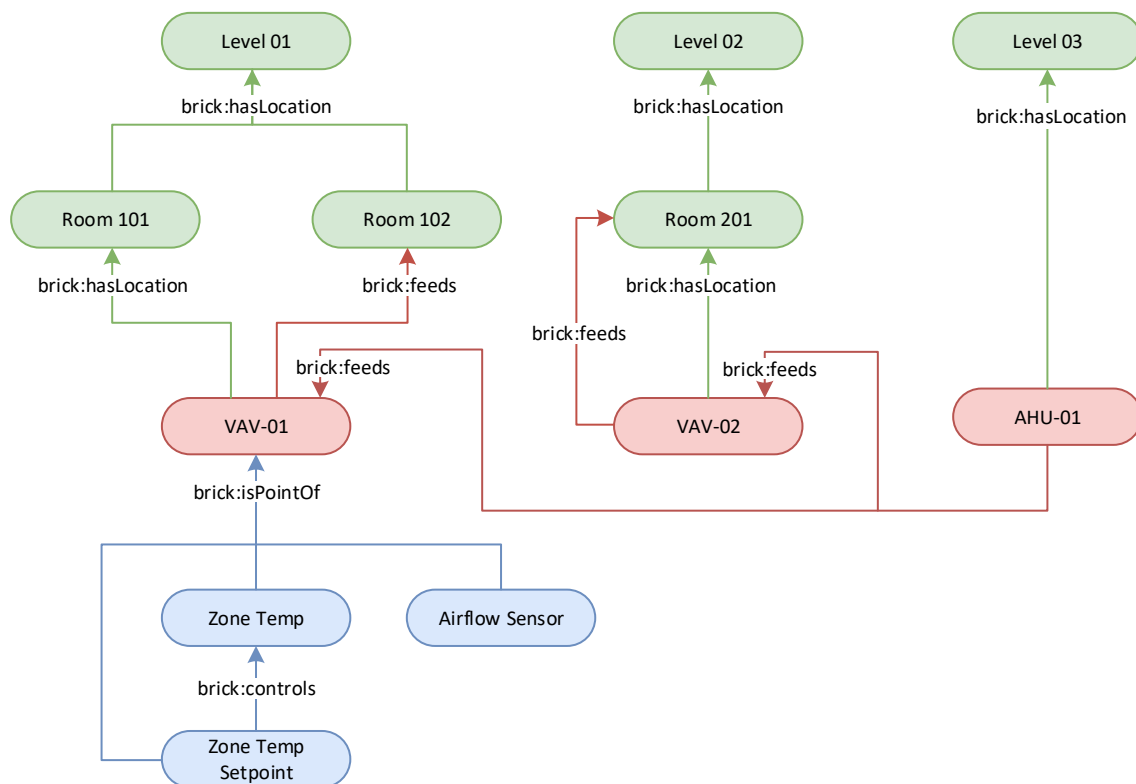


Figure 7 Brick AHU VAV Serves Relationship Example

The above figure 7 shows an example of part of a typical building VAV AHU system and how the Serves Relationships should be used in this scenario.

Note: The *brick:controls* relationship is not technically a Serves Relationship, however it can provide some additional and useful information about how the software and hardware points interact with one another.

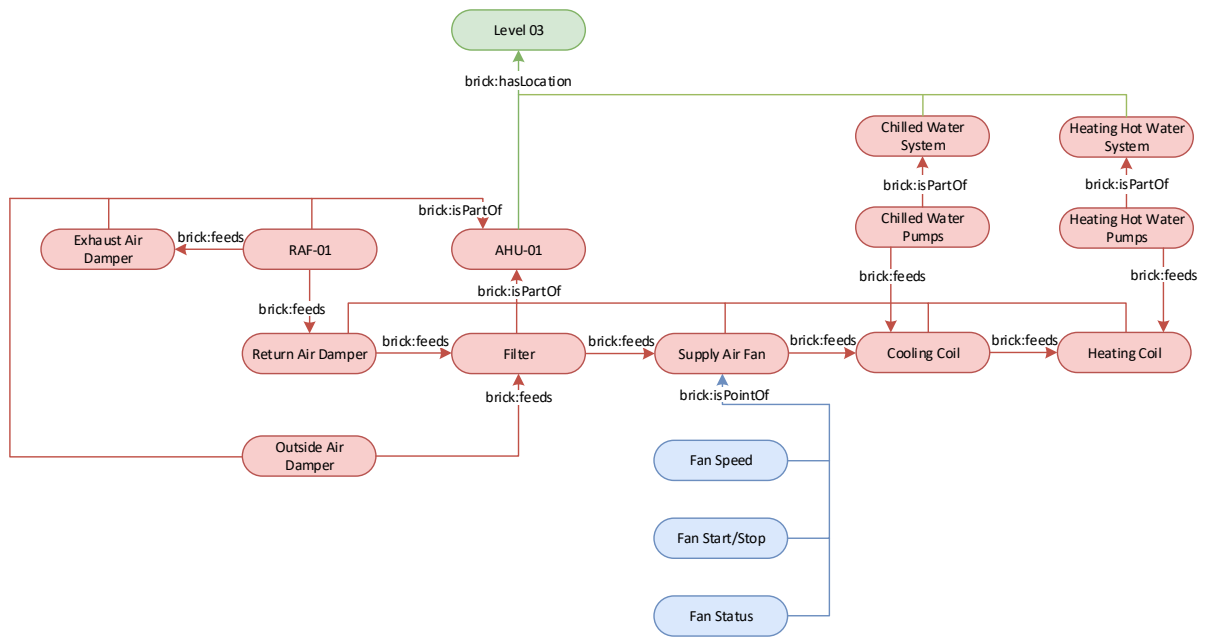


Figure 8 Brick AHU Serves Relationship Example

The above figure 8 shows an example of a typical AHU with associated sub-equipment such as Dampers, Filter, Return Air Fan, Supply Air Fan, Cooling and Heating Coils. With the correct use of flow relationships, the direction of airflow (or waterflow) through a piece of equipment can also be modelled.

I.e. The RAF feeds the EA & RA Damper, the RA and OA Damper feeds the Filter, the filter feeds the SAF, the SAF feeds the Cooling Coil and the Cooling Coil feeds the Heating Coil.

Haystack Minimum Required Relationships

“Ref” which is short for reference are special tags within haystack tagging models that refer to relationships between different entities.

Depending on the Ref being used, certain requirements need to be met for a Ref to be valid.

For Example, the “airRef” only makes logical sense between an entity with an “Air Output” tag and an “Air Input” Tag.

Refs are also directional. For Example

ahu → **airRef** → **vav** (“AHU feeds air” to VAV”)

is **not** the same as:

vav → **airRef** → **ahu** (“VAV feeds air to AHU”)

Relationship Domains

To ensure consistency of relationships between haystack models, the following constraints on relationship domains are proposed.

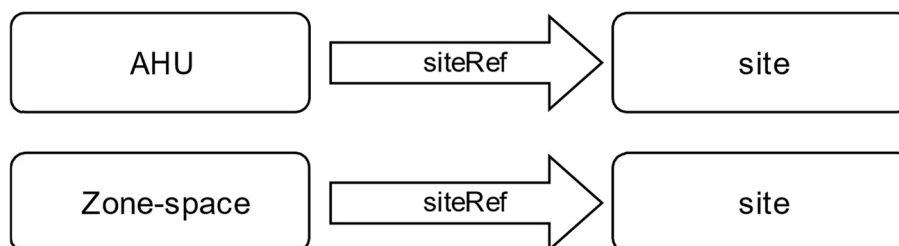
siteRef

<https://project-haystack.org/doc/lib-phIoT/siteRef>

Indicates an entity belonging to a site

From Tag	Relationship	To Tag
<ul style="list-style-type: none">• space• equip• point	siteRef	site

Example Use Cases:



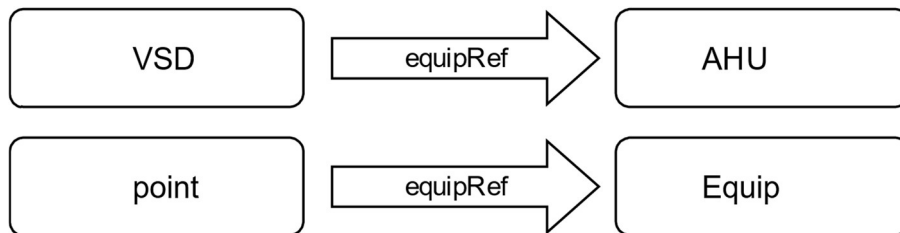
equipRef

<https://project-haystack.org/doc/lib-phloT/equipRef>

Indicates Containment of an Controller, Equipment or Point to an equipment entity.

From Tag	Relationship	To Ta
<ul style="list-style-type: none">• Controller• Equip• Point	equipRef	equip

Example Use cases



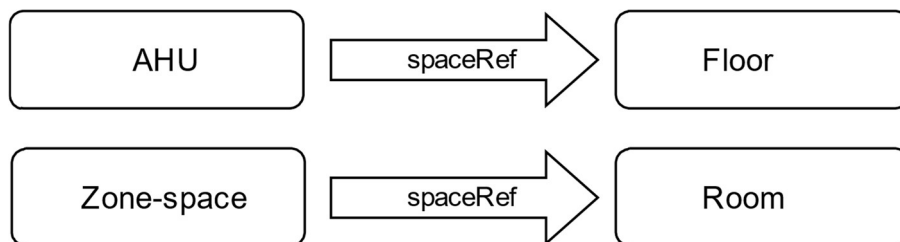
spaceRef

<https://project-haystack.org/doc/lib-phloT/spaceRef>

Indicates a space, equipment or point is located within a particular space

From Tag	Relationship	To Tag
<ul style="list-style-type: none">• space• equip• point	spaceRef	space

Example Use cases



airRef

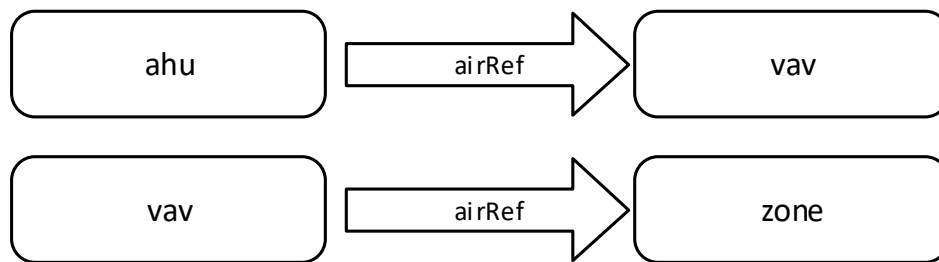
<https://project-haystack.org/doc/lib-phloT/airRef>

airRef Describes the flow of air from the source (air output) entities to the target (air input) entities.

air-input and **air-output** tags are typically required on air handling equipment types, such as vavs, ahus and HVAC-zones

From Tag	Relationship	To Tag
air-output	airRef	air-input

Example Use cases



Minimum Required Relationships

- All buildings (represented as site instances) must be located within a campus or portfolio instance. Linked via a *portfolioRef* reference.
- All floors must be located within a site instance. Linked via a *siteRef* reference.
- Each building must have at least 1 floor.
- Spaces such as rooms and zones within the building can only be associated to their appropriate floors. Linked via a *spaceRef* reference.
- All equipment must be associated with a floor or a space/location within a floor. Linked via a *spaceRef* reference.
- All points must be associated with an equipment and /or space/location. Linked via a *equipRef* if associated with an equipment and a *spaceRef* if associated with a space/location.

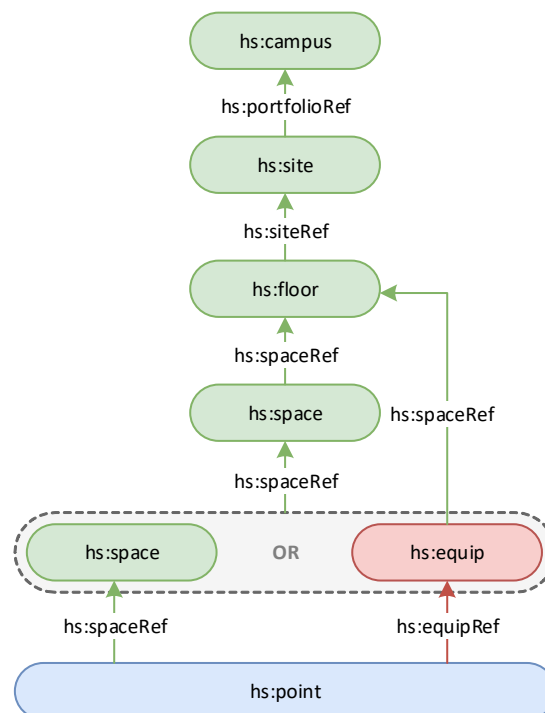


Figure 9 Haystack Minimum Relationship Requirements

Notes

- The “campus” marker is used in place of a site. Due to the lack of a “building” marker in haystack, a “site” marker is used to model a building.
- The “campus” and “portfolio” markers as well as the portfolioRef relationship are still only proposed new editions to Project Haystack and are currently not included in the latest Haystack definitions (version 3.9.12).
- In Haystack, siteRef relationships are flattened into all the child spaces, equipment (assets) and points. Only a siteRef relationship from the site entity to all of the associated floor entities is shown here. This is deliberate to maintain a hierarchical structure between the relationship tiers, similar to the space management modelling and Brick Schema explained earlier in the document.

Serves/Flow Relationships

Haystack Flow Relationship models must follow the following rules:

- Equipment (Assets) must be associated with the floor or space of where the equipment is located. Linked via a *spaceRef* reference.
- Equipment must be associated with the floor or space that it feeds. Generally linked via an *airRef* reference, but another reference could be used such as *chilledWaterRef*.
- Equipment must be associated with Equipment that it feeds. Linked via a any of the specific flow Ref references that is most applicable.
- Sub-Equipment must be associated with their parent Equipment. Linked via an *equipRef* reference.
- Points must be associated with their parent Equipment. Linked via an *equipRef* reference.

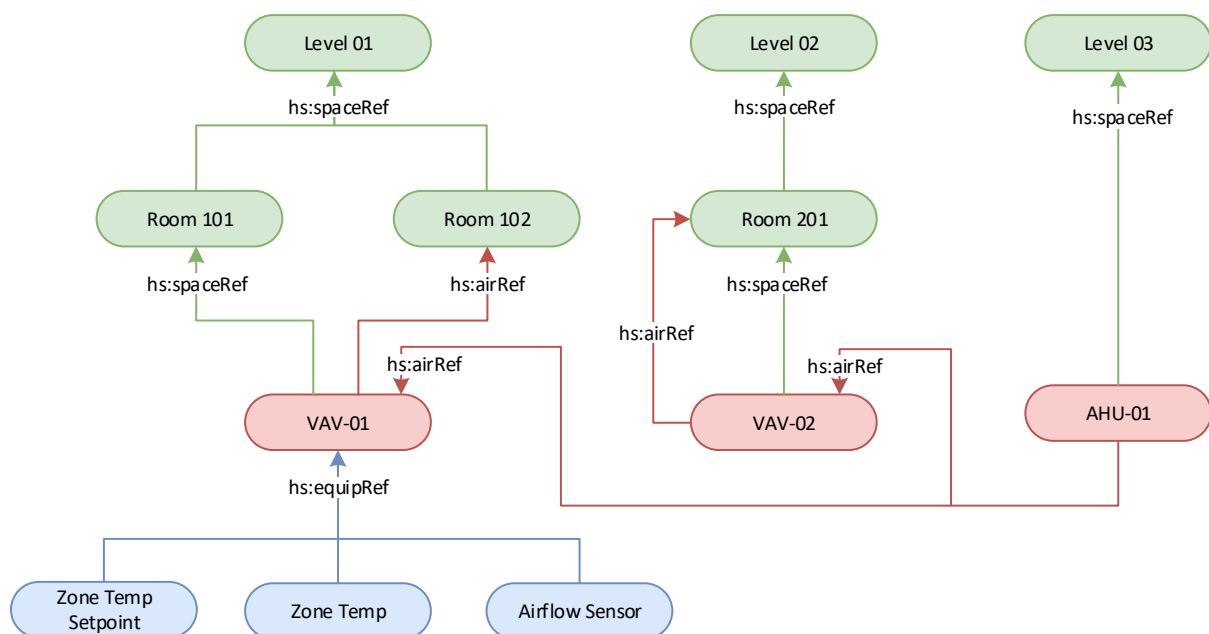


Figure 10 Haystack AHU VAV Serves Relationship Example

The above figure 10 shows an example of part of a typical building VAV AHU system and how the Flow Relationships should be used in this scenario.

Note: Haystack currently has no equivalent to the Brick Control Relationships.

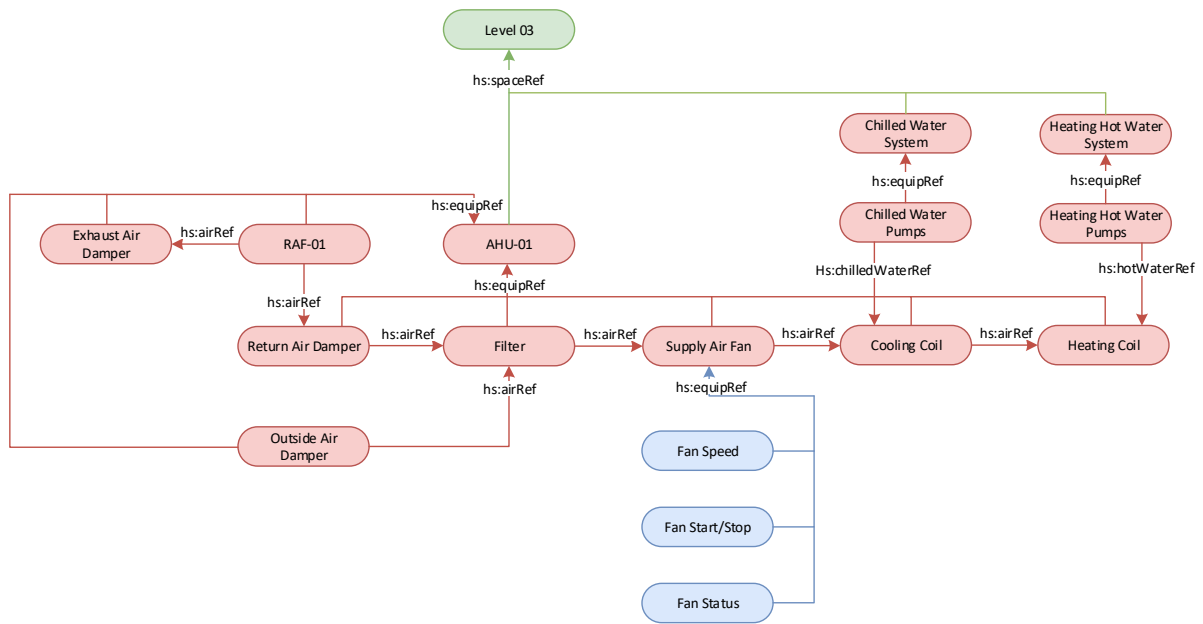


Figure 11 Brick AHU Serves Relationship Example

The above figure 11 shows an example of a typical AHU with associated sub-equipment such as Dampers, Filter, Return Air Fan, Supply Air Fan, Cooling and Heating Coils. With the correct use of flow relationships, the direction of airflow (or waterflow) through a piece of equipment can also be modelled.

I.e. The RAF has an airRef to the EA & RA Damper, the RA and OA Damper has an airRef to the Filter, the filter has an airRef to the SAF, the SAF has an airRef to the Cooling Coil and the Cooling Coil has an airRef to the Heating Coil.

BAS Tag Modelling

Brick Tagging Minimum Requirements

All Entities in a model must have the following tags at a bare minimum

URI

In addition to the Brick Equipment and Point Tags, it is a requirement that each Brick entity is tagged with a unique “uri” tag within the entire tag model as a whole. This is so that the tag entities within the space management system model can be referenced to tag entities within the BAS Brick Tag model. See beginning of document for information on the space management system and how it relates to the BAS Tagging Model.

To enforce strong URI formatting standards, all entity URIs must conform to the standard URI formatting

https://en.wikipedia.org/wiki/Uniform_Resource_Identifier

In most cases, it is strongly encouraged to include the closest related location Id followed by the equipment id for equipment and then the point id.

For example:

- campus1/bld10/lvl3/rm12/VAV_1/Zone_Temp
- campus1/bld10/roof/Chillers/Chiller_1

These URIs can be uniquely identified across an entire Asset portfolio, because they incorporate the unique location Id (from the space management system) as part of their URI

If the location codes were not part of the unique ID of the tagging model for a specific building,

For example:

- VAV_1/Zone_Temp
- Chillers/Chiller_1

These Tags will not be able to be incorporated into the tagging model, as some of these URIs could conflict with other URIs for any other building within the Asset portfolio.

Description

In addition to the Brick Equipment and Point Tags, a “label” tag should also be added to each Brick entity, which serves as a human readable descriptor when visualizing the model.

Equipment

All equipment (asset) entities must be tagged with the “Equipment” Tag and system category it belongs to such as:

HVAC, Lighting_System, Fire_System, Electrical_System, etc.

Each Equipment entity should also be identified as a type of equipment with a Tag such as:

VAV, AHU, FCU, Chiller, Boiler, etc.

Some examples of Equipment Entities and their associated Tags are as follows:

Equipment	Tags	Value	Type
Chiller	AbsoluteUri	campus1/bld10/basement/chiller1	string
	Label	Chiller 1	string
	Equipment		marker
	HVAC		marker
	Chiller		marker
Chilled Water Pump	AbsoluteUri	campus1/bld10/basement/chwp1	string
	Label	Primary Chilled Water Pump 1	string
	Equipment		marker
	HVAC		marker
	Pump		marker
	Chilled Water		marker
Boiler	AbsoluteUri	campus1/bld10/roof/boiler1	string
	Label	Boiler 1	string
	Equipment		marker
	HVAC		marker
	Boiler		marker
FCU	AbsoluteUri	campus1/bld10/lvl2/rm201/fcu01	string
	Label	Room 201 Fan Coil Unit	string
	Equipment		marker
	HVAC		marker
	FCU		marker
	Terminal_Unit		marker

Point

All Brick Point entities must be tagged with the “Point” Tag.

A point entity should also be tagged with exactly one of the following Point Function Tags:

- Alarm
- Sensor
- Command
- Setpoint
- Status
- Fault
- Parameter

This is to identify the point as either a sensor (input), command (output) and other internal control variable / software points.

Some examples of Point Entities and their associated Tags are as follows:

Point	Tags	Value	Type
Supply Air Temperature Sensor	absoluteUri	campus1/bld10/lvl2/rm201/fcu01/sat	string
	label	Fan Coil Unit Supply Air Temperature	string
	Point		marker
	Air		marker
	Sensor		marker
	Supply Temperature		marker
Supply Air Temperature Setpoint	absoluteUri	campus1/bld10/lvl2/rm201/fcu01/sat-sp	string
	label	Fan Coil Unit Supply Air Effective Setpoint	string
	Point		marker
	Air		marker
	Setpoint		marker
	Supply Temperature		marker
Fan Start/Stop Command	absoluteUri	campus1/bld10/lvl2/rm201/fcu01/fan-ss	string
	label	Fan Coil Unit Fan Start Stop	string
	Point		marker
	Fan		marker
	Command		marker
	On Off		marker marker
Fan Status	absoluteUri	campus1/bld10/lvl2/rm201/fcu01/fan-st	String
	label	Fan Coil Unit Fan Status	string
	Point		marker
	Fan		marker
	Status		marker
	On Off		marker Marker

Tag Groups

Tag groups are a set of predefined marker Tags designed specifically for most BAS standard equipment and point types.

Where possible, it is recommended that Tag Groups are used in place of individual Tags where an applicable Tag Group exists as it has all of the minimum required equipment and point tagging already defined within these Tag Groups.

Some examples of Brick Tag Groups are as follows:

Equipment	Tag Group	Predefined Tags
AHU	ahu	AHU, Equipment, HVAC
Chiller	chiller	Chiller, Equipment, HVAC
VAV	vav	VAV, Equipment, HVAC, Terminal_Unit

Point	Tag Group	Predefined Tags
Zone Temp	Zone_Air_Temperature_Sensor	Air, Air_Temperature_Sensor, Point, Sensor, Temperature, Temperature_Sensor, Zone, Zone_Air_Temperature_Sensor
Zone Temp Setpoint	Zone_Air_Temperature_Setpoint	Air, Air_Temperature_Setpoint, Point, Setpoint, Temperature, Temperature_Setpoint, Zone, Zone_Air_Temperature_Setpoint
Chilled Water Valve	Valve_Command	Command, Point, Valve, Valve_Command

Haystack Tagging Minimum Requirements

All Entities in a model must have the following tags at a bare minimum

Unique Id

All entities need to be uniquely identified via the “id” tag. This serves as a unique identifier across the entire haystack tagging model.

URI

In addition to the Haystack Equipment and Point Tags, it is a requirement that each Haystack entity is tagged with a unique “uri” tag within the entire tag model as a whole. This is so that the tag entities within the space management system model can be referenced to tag entities within the BAS Haystack Tag model. See beginning of document for information on the space management system and how it relates to the BAS Tagging Model.

To enforce strong URI formatting standards, all entity URIs must conform to the standard URI formatting

https://en.wikipedia.org/wiki/Uniform_Resource_Identifier

In most cases, it is strongly encouraged to include the closest related location Id followed by the equipment id for equipment and then the point id.

For example:

- campus1/bld10/lvl3/rm12/VAV_1/Zone_Temp
- campus1/bld10/roof/Chillers/Chiller_1

These URIs can be uniquely identified across an entire Asset portfolio, because they incorporate the unique location Id (from the space management system) as part of their URI

If the location codes were not part of the unique ID of the tagging model for a specific building,

For example:

- VAV_1/Zone_Temp
- Chillers/Chiller_1

These Tags will not be able to be incorporated into the tagging model, as some of these URIs could conflict with other URIs for any other building within the Asset portfolio.

Description (dis Tag)

Dis, short for “Description” Serves as a human readable name for Haystack Entities.

Equipment

All equipment (asset) entities must be tagged with the “equip” Tag.

They must also be identified as a type of equipment with a Tag such as:

vav, ahu, fcu, chiller, boiler, etc.

Some examples of Equipment Entities and their associated Tags are as follows:

Equipment	Tags	Value	Type
Chiller	id	campus1/bld10/basement/chiller1	string
	dis	Chiller 1	string
	equip		marker
	chiller		marker
Chilled Water Pump	id	campus1/bld10/basement/chwp1	string
	dis	Primary Chilled Water Pump 1	string
	equip		marker
	pump		marker
	chilled water		marker
Boiler	id	campus1/bld10/roof/boiler1	string
	dis	Boiler 1	string
	equip		marker
	boiler		marker
FCU	id	campus1/bld10/lvl2/rm201/fcu01	string
	dis	Room 201 Fan Coil Unit	string
	equip		marker
	fcu		marker
	airHandlingEquip		marker
air-output		marker	

Point

All point entities must be tagged with the “point” Tag.

A point entity must also be tagged with exactly one of the following Point Function Tags:

- sensor
- cmd
- sp

This is to identify the point as either a sensor (input), command (output) or setpoint (internal control variable / software point).

Some examples of Point Entities and their associated Tags are as follows:

Point	Tags	Value	Type
Supply Air Temperature Sensor	id	campus1/bld10/lvl2/rm201/fcu01/sat	string
	dis	Fan Coil Unit Supply Air Temperature	string
	point		marker
	air		marker
	discharge		marker
	sensor		marker
	temp		marker
Supply Air Temperature Setpoint	id	campus1/bld10/lvl2/rm201/fcu01/sat-sp	string
	dis	Fan Coil Unit Supply Air Effective Setpoint	string
	point		marker
	air		marker
	discharge		marker
	sp		marker
	temp		marker
Fan Start/Stop Command	id	campus1/bld10/lvl2/rm201/fcu01/fan-ss	string
	dis	Fan Coil Unit Fan Start Stop	string
	point		marker
	fan		marker
	cmd		marker
	run		marker
Fan Status	id	campus1/bld10/lvl2/rm201/fcu01/fan-st	string
	dis	Fan Coil Unit Fan Status	string
	point		marker
	fan		marker
	sensor		marker
	run		marker

Tag Groups

Tag groups are a set of predefined marker Tags designed specifically for most BAS standard equipment and point types.

Where possible, it is recommended that Tag Groups are used in place of individual Tags where an applicable Tag Group exists as it has all of the minimum required equipment and point tagging already defined within these Tag Groups.

Some examples of Haystack Tag Groups are as follows:

Equipment	Tag Group	Predefined Tags
AHU	ahu	<i>ahu, entity, equip, airHandlingEquip, output, air-output, input, elec-input</i>
Chiller	chiller	<i>chiller, entity, equip</i>
VAV	vav	<i>vav, entity, equip, airTerminalUnit, output, air-output</i>

Point	Tag Group	Predefined Tags
Zone Temp	zoneAirTempSensorPoint	<i>air, point, sensor, temp, zone</i>
Zone Temp Setpoint	zoneAirTempEffectiveSpPoint	<i>air, effective, point, sp, temp, zone</i>
Cooling Valve Command	chilledWaterCoolValveCmdPoint	<i>chilled, cmd, cool, point, valve, water</i>