Project Haystack

Implementing Project Haystack

Applying Haystack Tagging for a sample building

https://project-haystack.org/

Introduction

Document Goals

This document aims to introduce core Haystack data model concepts and solidify further understanding by demonstrating a simple real-world example.

Project Haystack Data Model

The Project Haystack semantic data model is used to represent the various equipment and relationships in automation, control, energy, HVAC, lighting, and other environmental systems. This standardised structure makes it easy to exchange information, and unlock value from the vast amounts of data being generated by the smart devices that permeate our homes, buildings, factories, and cities.

Background Reading

This document assumes that the reader has already become familiar with Project-Haystack.org and the basic concepts of semantic tagging at the conceptual level. The link below is a recommended primer for readers who are new

Introduction to Project Haystack

Core Data Model Concepts

Entities and Tags

Project Haystack maps physical objects in buildings to entities. You can think of an entity as an abstraction for a single Site (a building with an address), a piece of equipment in a building (e.g. HVAC unit, an electric meter, or a variable air volume box), or a sensor point (Digital or analog sensor that generate signals and data e.g .temperature sensor, pressure sensor, on/off switch, etc).

A tag defines a fact or attribute about an entity. For example, if we apply the site tag to an entity, then we are declaring that the entity represents a building that has an address as per the Haystack definition of the site tag.

Currently there are over 200 tags defined by Project Haystack that can be applied in various combinations to define virtually any set of descriptors or relationships for equipment, systems and associated data in a standardized way. The tagging method also means you only have flexibility to apply tags that are relevant to your application.

Site, **equip** and **point** are the 3 main tags used for an entity definition in Project Haystack. This document will show a simple example of how we organise them with other tags to entities in a simple building.

Tag Kinds (Tag Types)

Project Haystack has several tag types or kinds. In this document we will focus on three that are commonly used.

Marker

A marker tag is merely an annotation with no associated value. Marker tags are used to indicate a "type" or "is-a" relationships. For example, Applying the **ahu** tag to an entity indicates that it’s an “Air Handler Unit which heats and/or cools air”.

Marker tags can be optionally applied to an entity in any combination to provide fine-grained detail. For example, to describe an energy meter’s kWh value, we could add the following combination of marker tags:

- **point**: ✓
- **elec**: ✓

- energy: ✓

Note:

1. *In this example it is common for the entity to have other tags specified, e.g. dis, units.*
2. *The ✓ notation is used throughout this document to denote the tag is a Marker kind*

Str

A string (str) tag has an associated text value. Str tags are commonly used for human readable descriptions. For example, the **dis** tag is used for an entities display name 'Incoming Energy Meter', **geoAddr** is used for defining the street address '455 Larkspur Dr. Baviera, CA 92908'.

For some str tags, Project Haystack defines a list of permissible text values. For example, the **unit** tag is used for defining the unit of measurement, where the text value can only be set from a record in the unit database maintained by Project Haystack.

In the point entity listed above we could add the following str tags:

- **dis**: 'Energy'
- **unit**: 'kWh'

Ref

Ref tags are assigned values of other entities to help us define how entities within a data model relate to one another. For example, we can associate an **equip** entity to a particular **site** entity via **siteRef**, or associate a **point** to a particular **equip** via **equipRef**

In the **point** entity example above, we could add the **equipRef** tag and set it's value to another entity, which is the energy meter itself.

- **equipRef**: '@Id_of_Energy_Meter_Entity'

Note: '@Id_of_Energy_Meter_Entity' value simply identifies another ***equip*** *entity which is an Energy Meter.*

Example Summary

To summarise the above examples we would have described the Energy Meters kWh reading:

- **point**: ✓
- **elec**: ✓
- **energy**: ✓
- **dis**: 'Energy'
- **kind:** 'Number'
- **unit:** 'kWh'
- **equipRef**: '@Id_of_Energy_Meter'

Note: There are other common tags that are applied to points e.g. ***sensor, currVal****, and* ***his****. They have been omitted for simplicity.*

Other Tag Types

The remaining tag kinds used in Project Haystack are listed below:

- **Bin**: a binary blob with a MIME type formatted as Bin(text/plain)
- **Bool**: boolean "true" or "false".
- **Coord**: geographic coordinate in latitude/longitude formatted as C(lat,lng)
- **Date**: an ISO 8601 date as year, month, day: 2011-06-07.
- **DateTime**: an ISO 8601 timestamp followed by timezone name: 2011-06-07T09:51:27-04:00 New_York
- **NA**: singleton value which represents not available for missing data
- **Number**: integer or floating point number annotated with an optional unit of measurement.
- **Time**: an ISO 8601 time as hour, minute, seconds: 09:51:27.354.
- **Uri**: a Universal Resource Identifier.
- **XStr**: extended typed string which specifies a type name a string encoding

There are also three collection tag kinds:

- **List**: list of zero or more values of any type
- **Dict**: an associated array of name/value tag pairs
- **Grid**: a two dimensional table of columns and rows, see Grids

Defining a Sample Project – A Building in Gaithersburg MD, USA

Project Details

This example project is a Building in Gaithersburg and has the following characteristics:

- Address: 18212 Montgomery Village Ave, Gaithersburg, MD, USA
- Timezone: New York
- Building Size: 8013 sq ft
- Primary function: Retail store
- Year built: 2001

It also has the following Equipment Systems:

- Rooftop HVAC Units: RTU-1 and RTU-2
- Electrical Meters: Main meter, HVAC submeter, lighting submeter, a plug load submeter
- A Gas meter
- Lighting circuits: main lights and parking lights

Describing the Basics

Step 1: Defining the Entity for the Gaithersburg Site

Let’s start our exploration of Haystack tagging by showing how we would apply tags to an entity that defines the Gaithersburg building:

Here are the entries we would make to describe our site and its essential attributes:

Entity#	Tag Name	Value	Type	Explanation
1	id	@2180b666-430b2363	Ref	This is a unique id specific to the database storing the information. Haystack identifies the concept of an ID, but it’s up to the application

				storing the data to create that ID for its internal use.
	site	✓	Marker	This marker tag simply means the entity is a “site”
	dis	‘Gaithersburg’	Str	This tag holds the display name that will be associated with this Site

From here we can define further useful information by applying additional optional tags:

Entity #	Tag Name	Value	Type	Explanation
1	geoAddr	‘18212 Montgomery Village Ave, Gaithersburg, MD’	Str	The street address of the Site
	geoCity	‘Gaithersburg’	Str	The city the Site is located in
	geoCoord	C(39.154824,-77.209002)	Coord	Geo coordinates
	geoCountry	‘US’	Str	
	geoPostalCode	‘20879’	Str	
	geoState	‘MD’	Str	
	tz	‘New_York’	Str	This defines the time zone the site is located in according to the Haystack time zone definitions.

Our site entity is done! By applying the above tags to the entity representing the Gaithersburg site, we have captured the essential information about the site. This information can be read by external applications which are then able to interpret the meaning of the record from the tags.

Step 2: Defining an Equipment

Earlier we noted that our sample facility has the following equipment:

- Rooftop HVAC Units: RTU-1 and RTU-2
- Electric meters: Main meter, HVAC submeter, lighting submeter, a plug load submeter
- Gas meter
- Lighting circuits: main lights and parking lights

We will start the process of tagging our equipment by creating a record and the associated tags to describe one of the HVAC units.

Entity#	Tag Name	Value	Type	Explanation
2	id	@2180b666-7032054c	Ref	Again our record will have a unique id in the system it is being created in
	equip	✓	Marker	This is a marker tag indicating that this entity is an “equip” - a piece of equipment. That gives it a identifiable place in the Site-Equip-Point hierarchy.
	dis	‘RTU-1’	Str	

The next set of tags describe further facts about this entity using “marker” tags:

Entity#	Tag Name	Value	Type	Explanation
2	ahu	✓	Marker	This is a marker tag indicating that this entity is an “ahu” - an air handler.
	hvac	✓	Marker	This marker tag indicates that this equip is HVAC in nature (versus lighting for example)
	rooftop	✓	Marker	This marker tag indicates that this equip is a rooftop type ahu

Step 3: Defining Relationships Between Entities with “ref” Tags

Next we will add “ref” tags that define relationships between this record and other entities in our project:

Entity#	Tag Name	Value	Type	Explanation
2	siteRef	**@** 2180b666-430b2363 (Gaithersburg)	Ref	This ref indicates that this piece of equipment belongs to the site - Gaithersburg
	elecMeterRef	@2180b666-7032054d (Gaithersburg ElecMeter-Hvac)	Ref	This tag associates this ahu with a specific meter. It says that this ahu is a load measured by the entity with an id of @2180b666-7032054d named “Gaithersburg ElecMeter-Hvac”

Note: Haystack requires all tags to be explicit. They are not automatically applied or inherited when ref tags are used. Engineering tools for configuring Haystack systems can be designed to force a hierarchical relationship or automatically add tags as part of their functionality, but the Haystack standard expects explicit definitions. Haystack Filters however, can be used to walk or inspect hierarchies in the data model.

Step 4: Defining a “Point” Associated with our Equipment

To review, Haystack uses a very simple, minimal, relationship structure of Site/Equip/Point. We have defined our Site and an Equip, in the next section we will define a point that is associated with the Rooftop Air Handler equipment. We will use the discharge air sensor measuring the temperature of the supply air for our example.

Like the other entities, this one starts with an identifier for the record in the database:

Entity#	Tag Name	Value	Type	Explanation
3	Id	@218a0616-0b5e382b	Ref	The record will have a unique id in the system where it is being created

	dis	‘Discharge Air Temp’	Str	Name of the point
	point		Marker	Marker indicating a point entity

The following tags indicate what the point measures and the corresponding data attributes point’s value.

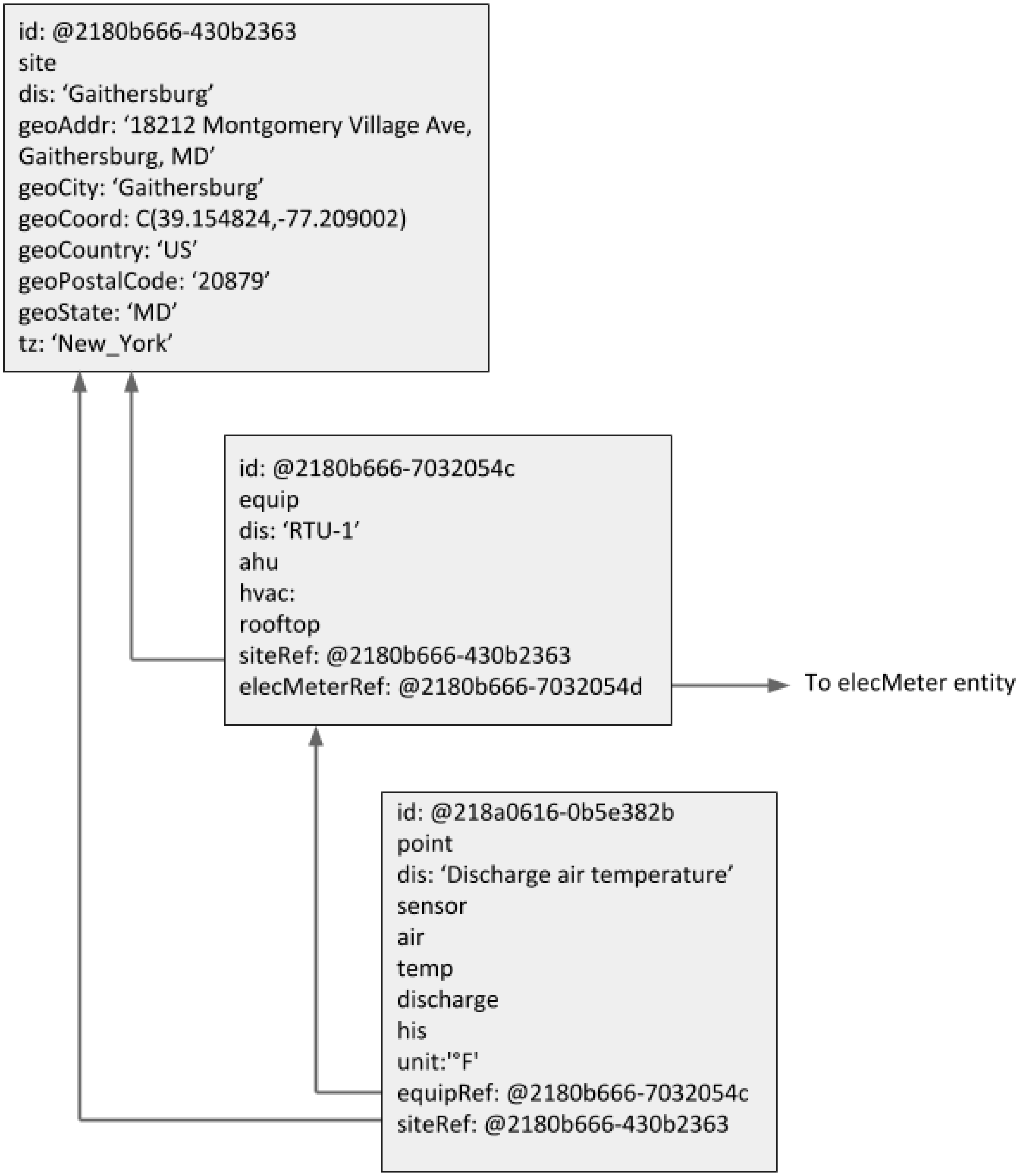
Entity#	Tag Name	Value	Type	Explanation
3	discharge		Marker	Marker indicating point location is discharge air
	air		Marker	Marker indicating point measures the medium air
	temp		Marker	Marker indicating point measures temperature
	sensor		Marker	Marker indicating point is a sensor
	his		Marker	Marker indicating point has history of timestamp/value pairs
	kind	‘Number’	Str	Indicates the data is expressed as a number
	unit	‘°F’	Str	Engineering unit representing the unit of measure

Finally we define tags that indicate the site and equipment where the point belongs

Entity#	Tag Name	Value	Type	Explanation
3	siteRef	**@** 2180b666-430b2363 (Gaithersburg)	Ref	This ref indicates this point belongs to the site - Gaithersburg
	equipRef	@2180b666-7032054c (RTU-1)	Ref	This ref indicates this point belongs to the equipment - Gaithersburg RTU-1

Putting it all together

After defining these 3 entities, we can visually represent the entities and relationships in the diagram below.



id: @2180b666-430b2363
site
dis: ‘Gaithersburg’
geoAddr: ‘18212 Montgomery Village Ave, Gaithersburg, MD’
geoCity: ‘Gaithersburg’
geoCoord: C(39.154824,-77.209002)
geoCountry: ‘US’
geoPostalCode: ‘20879’
geoState: ‘MD’
tz: ‘New_York’

id: @2180b666-7032054c
equip
dis: ‘RTU-1’
ahu
hvac:
rooftop
siteRef: @2180b666-430b2363
elecMeterRef: @2180b666-7032054d

To elecMeter entity

id: @218a0616-0b5e382b
point
dis: ‘Discharge air temperature’
sensor
air
temp
discharge
his
unit:'°F'
equipRef: @2180b666-7032054c
siteRef: @2180b666-430b2363

Additional point entities associated with our RTU equipment would be added in a similar way using the appropriate tags. These points might include:

- One or more stages of cooling that are modeled as on/off boolean points
- A discharge air pressure sensor
- A fan modeled as an on/off boolean point
- An outside air damper modeled as a two position boolean point
- A return air temperature sensor
- A zone temperature
- A zone temperature setpoint value

Inspecting the model and exchanging data

Now that we've modeled our equipment in a standardised method, we can now inspect and extract the information from a Haystack server in a streamlined manner.

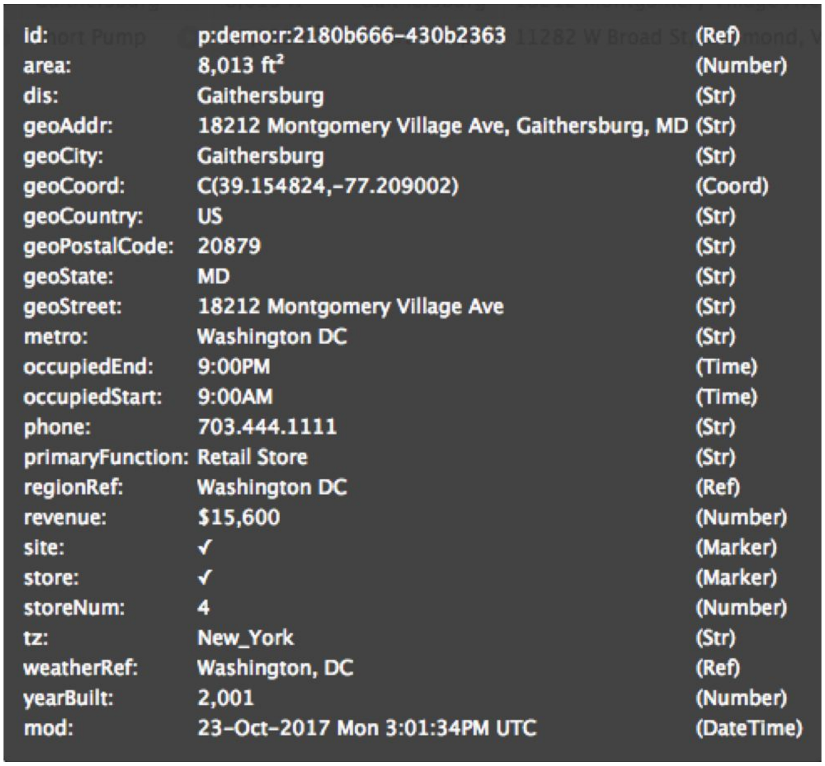
Viewing Entities

The information about our Project can be represented in many different ways. For example, we could capture the information in an Excel spreadsheet, a database record, or in the configuration files in a building automation controller. We've provided some examples of commercial software below to highlight this concept.

Note: The examples below include additional tags beyond the essential ones we created. Some are standard haystack tags, others are non-standard custom tags. Note that the Haystack methodology is extensible allowing you to create your own tags to capture information like: region, year built, revenue, etc. See [*custom tags*]() *for more on this subject.*

Example 1

The image below shows an application that displays all of the tag information for the Gaithersburg site in a pop up:



id:	p:demo:r:2180b666-430b2363	(Ref)
area:	8,013 ft²	(Number)
dis:	Gaithersburg	(Str)
geoAddr:	18212 Montgomery Village Ave, Gaithersburg, MD	(Str)
geoCity:	Gaithersburg	(Str)
geoCoord:	C(39.154824,-77.209002)	(Coord)
geoCountry:	US	(Str)
geoPostalCode:	20879	(Str)
geoState:	MD	(Str)
geoStreet:	18212 Montgomery Village Ave	(Str)
metro:	Washington DC	(Str)
occupiedEnd:	9:00PM	(Time)
occupiedStart:	9:00AM	(Time)
phone:	703.444.1111	(Str)
primaryFunction:	Retail Store	(Str)
regionRef:	Washington DC	(Ref)
revenue:	$15,600	(Number)
site:	✓	(Marker)
store:	✓	(Marker)
storeNum:	4	(Number)
tz:	New_York	(Str)
weatherRef:	Washington, DC	(Ref)
yearBuilt:	2,001	(Number)
mod:	23-Oct-2017 Mon 3:01:34PM UTC	(DateTime)

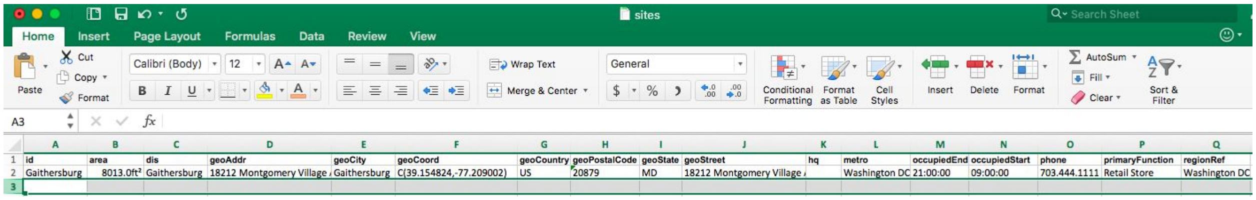
Example 2

Below is a CSV type representation of that same information:

```
id,area,dis,geoAddr,geoCity,geoCoord,geoCountry,geoPostalCode,primaryfunction,site,tz,
yearBuilt
@2180b666-430b2363 Gaithersburg,8013ft²,Gaithersburg,18212 Montgomery Village Ave,
Gaithersburg, MD,Gaithersburg,C(39.154824,-77.209002),US,20879,Retail
Store,✓,New_York,2001
```

Example 3

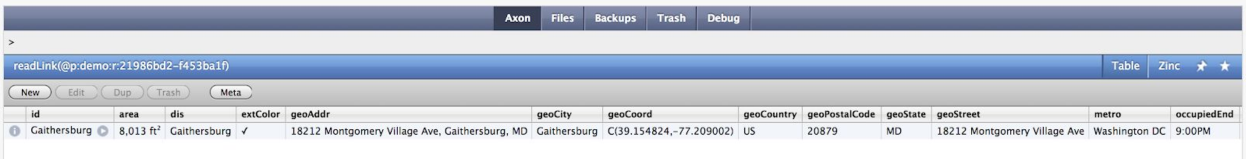
Here is that same tag information represented in Excel. Note that the tag names are column headers and the data is the value in the cell (note images is clipped on the right side):



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	id	area	dis	geoAddr	geoCity	geoCoord	geoCountry	geoPostalCode	geoState	geoStreet	hq	metro	occupiedEnd	occupiedStart	phone	primaryFunction	regionRef
2	Gaithersburg	8013.0ft²	Gaithersburg	18212 Montgomery Village A	Gaithersburg	C(39.154824,-77.209002)	US	20879	MD	18212 Montgomery Village A		Washington DC	21:00:00	09:00:00	703.444.1111	Retail Store	Washington DC
3																	

Example 4

Here is an example of the record and tags as a grid as presented in a commercial software application that uses Haystack tagging for defining entities (note image is clipped on the right side):

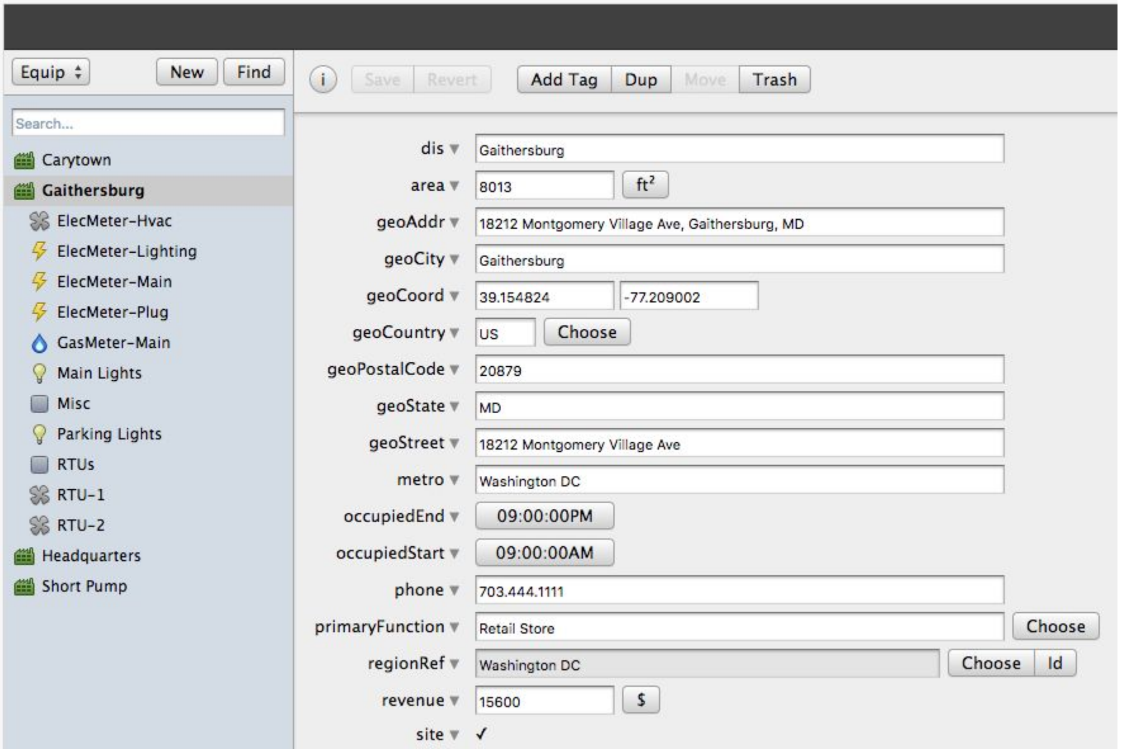


readLink(@p:demo:r:21986bd2-f453ba1f)

id	area	dis	extColor	geoAddr	geoCity	geoCoord	geoCountry	geoPostalCode	geoState	geoStreet	metro	occupiedEnd
Gaithersburg	8,013 ft²	Gaithersburg	✓	18212 Montgomery Village Ave, Gaithersburg, MD	Gaithersburg	C(39.154824,-77.209002)	US	20879	MD	18212 Montgomery Village Ave	Washington DC	9:00PM

Example 5

Finally, here is an example from a commercial software application that uses Haystack tagging for defining entities:



dis Gaithersburg
area 8013 ft²
geoAddr 18212 Montgomery Village Ave, Gaithersburg, MD
geoCity Gaithersburg
geoCoord 39.154824 -77.209002
geoCountry US
geoPostalCode 20879
geoState MD
geoStreet 18212 Montgomery Village Ave
metro Washington DC
occupiedEnd 09:00:00PM
occupiedStart 09:00:00AM
phone 703.444.1111
primaryFunction Retail Store
regionRef Washington DC
revenue 15600 $
site ✓

Finding Entities

Filters are a type of query language that are designed to find entities with particular tags of interest. This allows for manual inspection or automated software rules to look for types of equipment or relationships.

Example 1

Filter: ‘`meter`’

Returns: All entities with the **meter** marker tag. In our sample example above, it would return the Gaithersburg ElecMeter-Hvac entity id = @2180b666-7032054d.

Example 2

This is a complex example, but demonstrates how powerful the filters can be:

Filter: ‘*`point and temp and equipRef->ahu and siteRef->geoCity == "Gaithersburg"`*’

Returns: All entities with the following:

- Both point and temp marker tags;
- Contain an equipRef tag where the referenced entity also has an ahu marker tag; and

- Contain an siteRef tag where the referenced entity also has an geroCity tag that’s set to “Gaithersburg”

In our sample example above, it would return the “Discharge Air Temp point” entity, id = @218a0616-0b5e382b

Data Exchange

Equally important to data modeling, Project Haystack also standardizes how Haystack servers are to allow 3rd parties (owners, operators, or customers) to connect, inspect and exchange data via the REST API. With several open source tools available to ease the barrier of entry, all data in a Haystack server can be easily leveraged by any authorised person in a streamlined manner.

Data access types

Fundamentally the REST API supports 4 ways to exchange data:

1. Inspect a data model:
    a. Discover and learn via navigating entities
    b. Inspect via supplying a filter
    c. Read an entities known **id**
2. Retrieve and write historical time-series data
3. Watch data in real-time
4. Control points in real-time. For example, a remote control of a smart light.

Exchange Formats

Data is commonly exchanged from the REST API by two standardized formats zinc and json. Below is an example response from the above sample project.

Example 1 (ZINC Format)

The query requests the Haystack server respond to with all the entities (max 1000) that contain the **site** marker tag:

Request:

```
ver:"3.0"
filter,limit
"site",1000
```

Response:

```
ver:"3.0"
id,area,dis,geoAddr,geoCity,geoCoord,geoCountry,geoPostalCode,primaryfunction,site,tz,
yearBuilt
@2180b666-430b2363,8013ft²,"Gaithersburg","18212 Montgomery Village Ave, Gaithersburg,
MD","Gaithersburg",C(39.154824,-77.209002),"US","20879","Retail Store",M,New_York,2001
```

Further Information

Other Example Tags

With the background of our example, it may now be beneficial to look at a few examples from the current list of standard tags supported by Haystack.

Note: Numerous Working Groups are currently focusing on different applications and equipment systems, are adding to the list on a regular basis.

- **ac**: Indicates an alternating current (AC) electrical quantity or device.
- **ahu:** Air Handler Unit which heats and/or cools air.
- **ahuRef:** Associate an entity such as a vav with an ahu.
- **air:** Point associated with the measurement or control of air.
- **area:** Floor area of a site measured in ft^2 or m^2.
- **boiler:** Boiler equip used to generate hot water or steam for heating.
- **centrifugal:** Indicates chiller compressor which uses a continuous flow of fluid through an impeller.
- **cloudage:** The fraction of the sky obscured by clouds.
- **cmd:** Classifies a point as an output, AO/BO, command, or actuator.
- **co:** Carbon monoxide level point measured in "ppm".

The full list can be found at this link: https://project-haystack.org/tag

Custom Tags

The Haystack methodology is also extensible allowing you to create your own tags to capture information like: region, year built, revenue, etc.

For example, In our sample building if we wanted to link the “RTU-1” entity to a 3rd party asset maintenance system. We could define a custom str tag called **assetId**, and assign it a value that links it to the maintenance system e.g. ‘GB-HVAC-RTU-1’. This **assetId** is now searchable using Haystack filters, and allows for tighter integration between the two systems.

Reference Models

Example of Tagging Equipment

This work was performed by and for the State of Utah which has graciously offered to contribute it back to the Haystack Community for educational purposes. It provides a detailed tagging model covering building systems including those listed below:

- AHU
- FCU
- VAV
- Chiller
- Cooling Tower
- Boiler
- Fan
- Pump
- Electric
- Gas
- Domestic Hot Water

State of Utah Haystack Tagging Reference Model Example

Note: That this is an active project and may be modified over time.

Interactive Examples

This website below demonstrates visual examples of Project Haystack data models. A visual network library with a physics engine has been used to model Haystack entities and relationships. This helps to illustrate more complex models and how they relate.

Project Haystack Visual Examples

Note:

- *This website and example models are not representations of commercial Project-Haystack implementations. They are to simply illustrate complex structures for training purposes.*
- *Sections of these examples contain non-standard haystack tags.*