

# Guide Specification for Data Modeling of Building Systems and Equipment (Based on Project-Haystack.org Open-Source Data Modeling Standard)

v2021-12-30

**1.0 Purpose:** The purpose of a data modeling standard is to provide a consistent, standardized methodology for defining descriptive information, known as metadata, to describe the meaning and contextual relationships of data items associated with facility automation systems, equipment systems, energy metering systems, and other smart devices.

**1.1 Background:** Automation systems, equipment systems and smart devices make it possible to collect vast quantities of data including environmental conditions, equipment operational status, energy usage and performance metrics. The reality, however, is that these data typically exist in a low-level ad-hoc format without standard or consistent organization, making it difficult to interpret trends, perform analysis or generate useful reports and visualizations without significant manual effort. The result is large volumes of unstructured data, that requires substantial manual effort in order derive value from the data.

The first step in transforming smart device data into actionable intelligence is to give the data "context" so that we know exactly what each piece of data "means" and how fits into an overall system. For example, to compare the discharge air temperature of an Air Handling Unit (AHU) against the return air temperature, we must first "find" these two pieces of information. In most existing systems, the only indication of what a data point means is found by deciphering an arbitrary name the system integrator gave it during configuration such as "DA\_TEMP" to represent Discharge Air Temperature.

Given that there has previously not been any accepted standard for naming conventions associated with building systems and equipment, one key challenge the industry faces is to establish a common vocabulary to give meaning to information collected by the various building systems. Looking at the challenge from the in relation to the realities of complex buildings systems, it becomes apparent that it is not possible to capture the full extent of descriptive information desired simply using data point names, even if standardized. For example, to model a complete air distribution system to determine which AHUs feed which Variable Air Volume (VAV)s; or to know all the relationships between sub-meters and equipment loads in an electrical system. Point names that attempted to capture this range of information would be impossibly long, therefore structured combinations of descriptors are needed.

Project Haystack defines a methodology for defining this semantic information based on the concept of data "tagging", where a common vocabulary of tags for building equipment systems and smart devices is used to "markup up" equipment entities with meta data to describe their meaning and attributes, and to define ontological relationships between equipment systems. Application of this methodology enables the meaning and inter-relationships of the data items to be interpreted automatically by software applications, as well as human users. This enables owners, operators, manufacturers and service providers

to more efficiently utilize and derive value from the vast amounts of data smart systems produce.

Project Haystack methodologies, vocabulary and ontological definition techniques encompasses the entire value chain of building systems and related intelligent devices. The standard enables owners and consultants to specify utilization of these conventions to ensure a standards-based data model providing easier integration with external applications. In addition, the standard facilitates the “mapping” of Haystack semantic models with other relevant standards.

## 2.0 Technical Overview

**2.1 Overall Design Concepts:** The building equipment systems and/or automation systems controlling them shall employ a meta-model methodology based on the broadly accepted concept of “tags” to capture and represent semantic information as described below.

**Tags:** Tags are name/value pairs, associated with entities like AHUs, electric meters, etc. Tags are simple and dynamic, add structure, and provide the flexibility needed to establish standardized models of diverse systems and equipment and their ontological relationships. Tags shall allow easy customization of data models on a per-task, per-project or per-equipment basis, while retaining the ability to be interpreted by external applications by adhering to the standard, defined tag definition methodology and existing vocabulary. Tags shall support the definition of the following essential data elements as a minimum:

**Entity:** An Entity is an abstraction for a physical object in the real world. Examples of entities include sites, facilities, equipment, sensor points, weather stations, etc. In software-based systems, an entity might be modeled as a record in a database, an object in a building automation system, or a row in a csv file or spreadsheet.

**Id:** The id tag defines the unique identifier of an entity in system. A primary use is cross referencing entity relationships by means of a using a [Ref](#) (reference) value type. Ref value types are determined by individual application. For example, a “siteRef” Defines a reference to and site which contains a specific entity. The scope of an entity may be undefined, but must be unique within a given system or project. This identifier may be used by other entities to cross-reference entities, associations, and systems.

**Dis:** The dis tag is used with entities to define display text used to describe an entity. Dis values are intended to be short (less than 30 or 40 characters), but fully descriptive of the entity for a human user.

Additional tags are used to describe facts, associations and other characteristics of the data items associated with the equipment systems. All tags follow a fixed set of data types called Kinds which are shown below:

**Tag Kinds.** The tagging standard shall provide the following permitted tag value types:

- Marker: this tag type is merely a marker annotation. Marker tags are used to indicate a "type" or "is-a" relationship.
- Bool: boolean "true" or "false".
- Number: integer or floating point number annotated with a Unit of Measurement, where ideally, units of measure are prescribed for various tasks.
- Str: a string of Unicode characters.
- Uri: a Universal Resource Identifier.
- Ref: reference to another entity. The Project Haystack specification does not currently prescribe specific identities or reference mechanisms, but should be used to cross link entities. Refs are formatted with a leading "@" and require a specific subset of ASCII characters be used: a-z, A-Z, 0-9, underbar, colon, dash, or dot.
- Bin: a binary blob with a MIME type formatted as Bin(text/plain)
- Geo: Geographic identifiers such as an address or latitude/longitude
- Date: an ISO 8601 date as year, month, day: 2011-06-07.
- Time: an ISO 8601 time as hour, minute, seconds: 09:51:27.354.
- DateTime: an ISO 8601 timestamp followed by timezone name:  
2011-06-07T09:51:27-04:00 New\_York  
2012-09-29T14:56:18.277Z UTC

**Example of Usage.** The following presents an example for an entity describing a site:

```
id: @whitehouse
dis: "White House"
site
area: 55,000 ft2
geoAddr: "1600 Pennsylvania Avenue NW, Washington, DC"
tz: "New_York"
weatherStationRef: @weather.washington
```

The example presents an entity with seven tags: id, site, dis, area, geoAddr, tz, and weatherStationRef. By convention, when writing examples each tag will be listed on their own line or separated by a comma. The site tag has no explicit value, it is an example of a marker tag. The dis, geoAddr, and tz tags have string values, many which may be standardized by others, typically indicated by double quotes. The area tag has a number value indicated by a scalar with unit of square feet or square meters. The weatherStationRef tag is a reference to another entity, typically indicated using the "@" character.

**2.2 Standard Library of Tags and Library Extensibility:** The data modeling standard shall provide a comprehensive library of standard tags to address common equipment, building systems, and devices types. The standard vocabulary of tags shall be developed by an open-source community through means of an open discussion forum and working groups to enable industry experts and interested parties to discuss, submit, fine-tune and consensus-approve additional tags or standard schemas to address equipment, systems, and applications of numerous types. The open forum process shall be transparent to enable

continued development of a vocabulary and taxonomy that will enable semantic understanding of facilities engineering data across and outside of the industry.

**2.3 Support for Use of Custom Tags:** The semantic tagging methodology shall also allow for the creation of custom tags to meet unique application needs. The methodology shall provide a standard approach for defining custom tags to address application needs of data models on a per-task, per-project or per-equipment basis, while retaining the ability to be interpreted by external applications by adhering to the standard, defined tag definition methodology and vocabulary structure.

**2.4 Modeling of Ontological Relationships:** The semantic modeling methodology shall support the concept of Associations to define ontological relationships between entities. An [association](#) is a tag used on to cross-reference related entities via symbols. The term Association to denotes an entity-to-entity relationship versus instance data [relationships](#), which utilize the concept of Ref (Reference) tags.

Full technical details on the use of Associations can be found at: <https://project-haystack.org/doc/lib-ph/association>

### 3.0 Open API for Integration with Software Applications

The data modeling standard shall provide a documented HTTP-based, Application Programming Interface (API) to define a simple mechanism to exchange “tagged” data over web services.

Servers supporting the API shall be programmed to implement a set of “ops” or operations in accordance with public API documentation. An operation is a uri that receives a request and returns a response. Standard operations are defined to query databases, setup subscriptions, or read/write histories of time-series data. Operations are pluggable so vendors can enhance open REST interfaces with customized, value-added functionality for their own business purposes.

Both requests and responses are modeled as grids. Grids are encoded using standard Multipurpose Internet Mail Extensions (MIME) types for grid serialization, may be pluggable using HTTP content negotiation, and other standardized web service protocols.

Technical details on the Haystack HTTP API can be found at this link: <https://project-haystack.org/doc/docHaystack/HttpApi>

### 4.0 Requirements for Application of Semantic Tagging in Projects

The goal of the use of a data modeling standard is to ensure consistent modeling of building systems, devices and associated data. The following requirements outline the use of the modeling standard in applications related to buildings, energy, and facility management.

**4.1 Minimum Data Modeling Requirements:** The data tagging implementation shall utilize consensus approved data modeling tag definitions wherever possible to create a coherent

model of all equipment systems, devices and their associated data points. The current library of standard tags shall be in accordance with: <https://project-haystack.org/doc/index>

Additional user-defined tags may be used to add additional descriptive information to equipment systems beyond that available from the standard library called out in the above link.

As a minimum the following hierarchy of semantic relationships shall be provided when used in facilities-oriented applications:

**Site:** Building or other type of facility with unique street address. Including display name, description, size (area) as a minimum. References to Internet-available weather stations for Sites are highly recommended, as are creating tags to represent other relevant characteristics of a Site such as year constructed, facility usage type, occupancy class, schedule(s) of operation, building systems type (e.g., packaged or central HVAC).

**Equipment:** An equipment asset such as an Air Handling Unit (AHU), VAV Box, meter, etc. Tags shall include associations with sites via id reference and display name as a minimum. Use of tags to specify equipment and vendors, model numbers, year of installation, and similar descriptive meta data are recommended.

**Points:** Points are entities, typically associated with equipment, which produce data such as a sensor. Minimum tags for Points shall include: Ref IDs to associate points with their related sites and equipment, type of value (Boolean or numeric), units of measure, and other relevant descriptors. Such as ranges of acceptable values are recommended.

**4.2 Exposing the modeled data via HTTP API:** Software and web service applications, including control system devices will expose the model definitions described above using the Project Haystack HTTP API published as part of the Project Haystack standard, openly accessible and kept up to date at this link: <https://project-haystack.org/doc/docHaystack/HttpApi>

**4.3 Software Reference Implementations:** The standard shall provide reference implementations in Java and other common programming languages providing sample code for implementation of the HTTP API in software applications.

**4.4 Open Source Modules for Commercially Available Products.** The Project Haystack Community has developed, and makes available, a comprehensive implementation of the Haystack protocol in the form of a software module for use with NiagaraAX-based systems. The module, known as NHaystack, is licensed under the Academic Free License ("AFL") v. 3.0. Public access to the NHaystack software module shall be maintained via the [project-haystack.org](https://project-haystack.org) site.

## 5.0 Open Source Licensing Terms

The Data Modeling Standard for Smart Device Data modeling methodology, standards, supporting documentation and reference implementations shall be available via an open-source license at no cost.

**5.1** The open source license shall use the Open Source Initiative Academic Free License 3.0 model. Full details on the terms of the license are available at: <http://project-haystack.org/doc/License> and <http://opensource.org/licenses/AFL-3.0>

## **6.0 Technical Documentation**

Full technical information on the Project-Haystack Data Modeling Standard can be found at: <https://project-haystack.org/>

END OF DOCUMENT