



NS

18

**NIAGARA
SUMMIT**

**CONNECTING
THE WORLD**



Data Modeling

James Johnson

Objectives

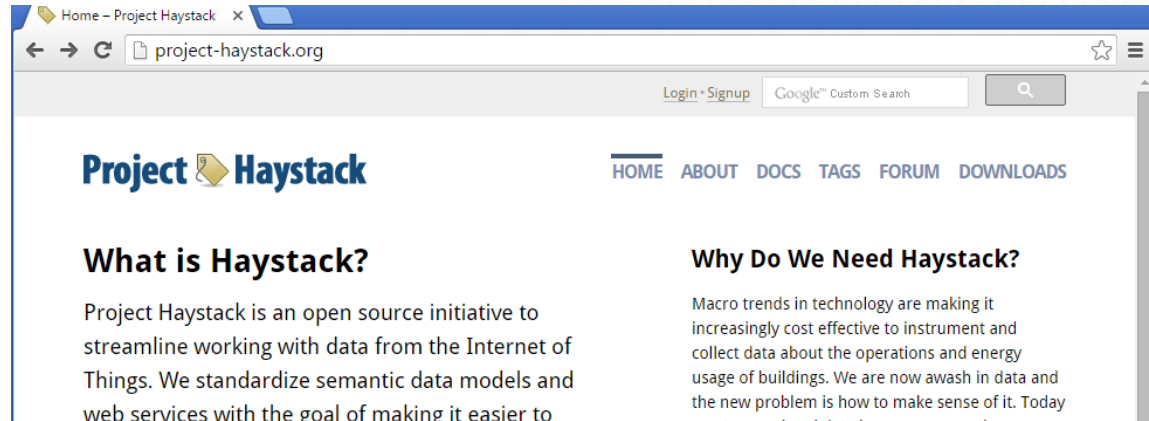
- Data modeling overview and terms
- Project Haystack
- Tag dictionaries
- Tags and relations
- Tag rules
- Using search and hierarchies

Data Modeling Overview and Terms

- **Metadata** – descriptive tags which provide context to either **define the structure or properties** of the data, often referred to as **Data about Data**.
- **Dictionary** – in **data modeling** defines the various **tags and relationships between entities** in the application.
- **Tagging** – the process of **applying metadata to entities** in the station.
- **Niagara Entity Query Language (NEQL)** – a simple language to **query data** from the station **using applied metadata**.
- Niagara 4 features which utilize data modeling include **Search, Hierarchies, Templates, Px views and Analytics**.

Project Haystack

- Open source initiative to streamline working with data from the IoT.
- Project specifications include a dictionary which defines the tag taxonomy for building systems such as networks, energy, HVAC and electrical distribution.
- Niagara 4 includes a Haystack dictionary.



Tag Dictionary Service

- Container for all **tag dictionaries** used in the Niagara 4 station
- Has an optional **Default Namespace Id** property





TagDictionaryService

Display Name	Value
<input type="radio"/> Status	<input type="text" value="{ok}"/>
<input type="radio"/> Fault Cause	<input type="text"/>
<input type="radio"/> Enabled	<input checked="" type="checkbox"/> true
<input type="radio"/> Default Namespace Id	<input type="text" value="n"/>
<input type="radio"/> Tag Rule Index Enabled	<input checked="" type="checkbox"/> true
<input type="radio"/> Indexed Tags	<input type="text" value="hs:ahu"/>
<input type="checkbox"/> Niagara	Niagara

Tag Dictionary

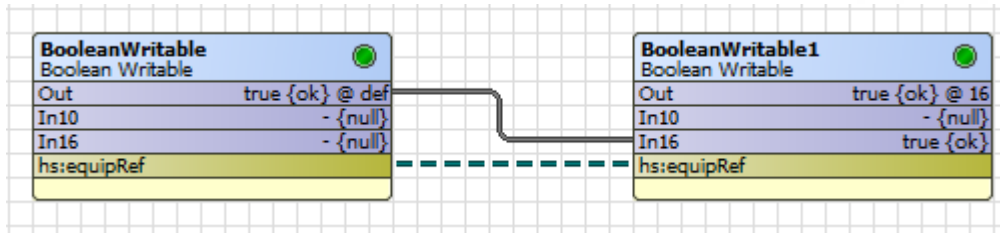
- Container for all **tag, tag group and relation definitions**.
- Optionally contains a list of **tag rules** which automatically apply tags and relations to entities in the station.
- Namespace is a shorthand reference to the dictionary which is usually 1 or 2 characters such as 'n' for Niagara or 'hs' for Haystack.

Niagara

Display Name	Value
<input type="radio"/> Status	{ok}
<input type="radio"/> Fault Cause	
<input type="radio"/> Namespace	n
<input type="radio"/> Enabled	<input checked="" type="checkbox"/> true
<input type="radio"/> Frozen	<input checked="" type="checkbox"/> true
▶  Tag Definitions	Tag Info List
 Tag Group Definitions	Tag Group Info List
▶  Relation Definitions	Relation Info List
▶  Tag Rules	Tag Rule List

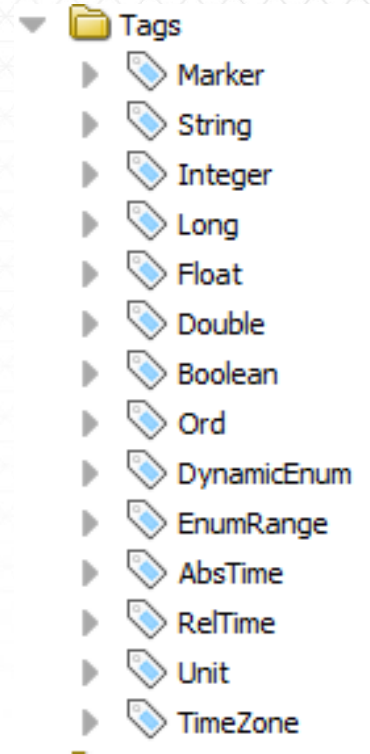
Relations

- Define how **entities** are **associated** to each other.
- Defined in the tag dictionary.
- **Links** are a special type of **relation** which allows data to flow between components.
- Can be viewed in the wiresheet, relation sheet or spy remote view.



Tags

- Defined in a dictionary and provide **semantic meaning** for that specific dictionary.
- **Marker** tags have no value, rather they apply some semantics by the fact they are applied.
- **Value** tags include additional semantic information such as a string, number, boolean or time value.

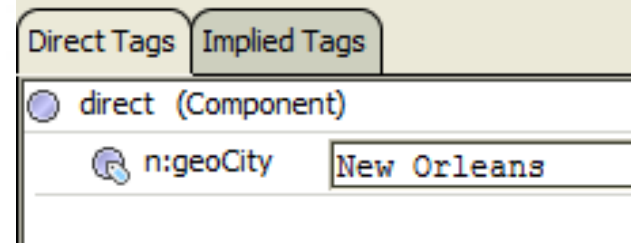


Direct Tags

- Typically assigned manually using edit tags dialog on the component or through batch editor.
- Added as **dynamic slots** on the component
- Slot name is the **escaped fully qualified name** of the tag

<TagDictionaryNamespace>:<TagName>

n:geoCity → b\$3ageoCity



Property	78	ws\$3ageoCity	ws\$3ageoCity	Dynamic		baja.ws\$3ageoCity
Property	79	n\$3ageoCity	n:geoCity	Dynamic	m	baja:String

Tag Rules and Implied Tags

- Rules can have complex conditions based on marker tags, value tags such as name, type, etc or relations.
- Rules can apply tags and relations.
- Tags and relations are implied at runtime.

Direct Tags	Implied Tags
n:displayName	SpaceTemp
n:type	control:NumericPoint
n:ordInSession	station: h:897b
n:station	BuildingJACE
n:point	marker
n:input	marker

Tag Rules

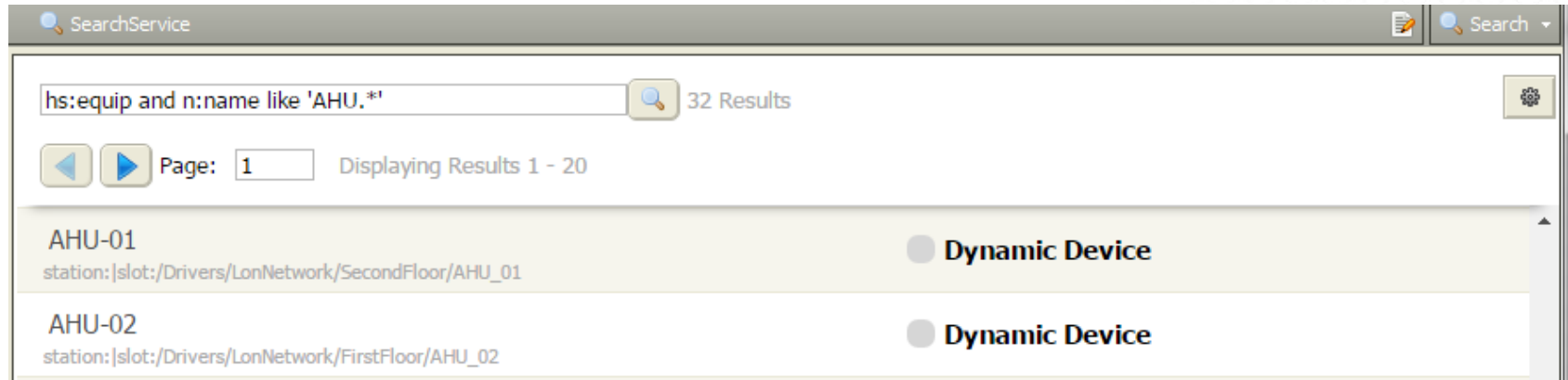
Display Name	Value
object tags	object tags
component tags	component tags
network tags	network tags
device tags	device tags
point tags	point tags
Condition	Is control:ControlPoint
Object Type	control ControlPoint
Tag List	Tag Info List
point	Marker
history	History Id Tag
input	Input Tag
output	Output Tag
Tag Group List	Tag Group Info List
Relation List	Relation Info List
schedule tags	schedule tags

Niagara Entity Query Language (NEQL)

- Provides a simple mechanism for **querying tagged entities** in a Niagara application.
- Query filters on **marker** and **value tags**.
- **Support traversing relations** defined on entities in a Niagara application.
- **Does not support BFormat** syntaxes.
- **Does not support tree semantics** nor **path statements** such as parent.parent.name or out.value
- Primary uses include **search service, hierarchies, analytics, templates** and **Px views**.

Search Service

- Utilizes both NEQL and BQL queries.
- Results display slot path and current value.
- Hyperlink to component, access to actions and hyperlink to web chart if applicable.

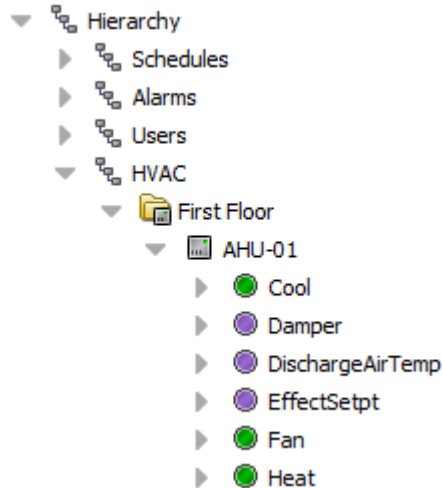


The screenshot displays the SearchService application interface. At the top, the title bar reads "SearchService" and includes a search icon and a "Search" dropdown menu. Below the title bar, a search input field contains the query "hs:equip and n:name like 'AHU.*'", followed by a search icon and the text "32 Results". Below the search field, there are navigation arrows and a "Page: 1" indicator, with the text "Displaying Results 1 - 20". The main content area shows two search results:

AHU-01 station: slot:/Drivers/LonNetwork/SecondFloor/AHU_01	<input type="radio"/> Dynamic Device
AHU-02 station: slot:/Drivers/LonNetwork/FirstFloor/AHU_02	<input type="radio"/> Dynamic Device

Hierarchies

- Efficient method of creating logical navigation trees.
- Leverages tags and relations using NEQL queries.
- Dynamically updates.
- Alternative to using nav files



HierarchyService (Hierarchy Service)	
Status	{ok}
Fault Cause	
Enabled	<input checked="" type="checkbox"/> true
Hierarchy Timeout	00000h 00m 45s [1ms - +inf]
Schedules	Hierarchy
Alarms	Hierarchy
Users	Hierarchy
HVAC	Hierarchy
Query Context	>> ↺
Status	{ok}
Fault Cause	
Scope	Hierarchy Scope Container
Tags	Hierarchy Tags
Cache Status	Not Cached
Cache Creation Time	null
Cache On Station Started	<input type="checkbox"/> false
Floors	Query Level Def: b:BldgFloor
Equipment	Relation Level Def: in: b:FloorRef
Points	Relation Level Def: out: n:childPoint

Templates and Px Views and Analytics, Oh My!

More sessions to come focusing on features which leverage data modeling

- 1:00 PM Today – Accelerated Engineering Using Templates
- 1:00 PM Tomorrow – From the Big Screen to the Small Screen
- 2:15 PM Tomorrow – Supersize My Supervisor
- 3:30 PM Tomorrow – Niagara Analytics

Summary

- Tag dictionaries define tags and relations used for semantic modeling.
- Tag rules automatically apply tags and relations to components in the station.
- The batch editor can be used to assign tags and relations in a batch fashion.
- NEQL is used to query entities in the station based on tags and relations.
- Hierarchies, Search, Templates, Px Views and Analytics all leverage NEQL and data modeling.

Questions

