



Vertex

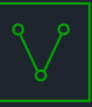
# Synapse Bootcamp

Module 9

Pivoting and Traversal in Storm

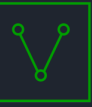
---

v0.4 - May 2024

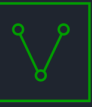


# Objectives

- Understand what it means to pivot in Synapse
- Use Storm to pivot through data
- Leverage specialized pivots such as wildcard pivots and joins
- Understand what it means to traverse ("walk") edges in Synapse
- Use Storm to traverse through data
- Leverage combined "pivot and walk" operations
- Understand how pivoting and traversal relate to Explore in the UI

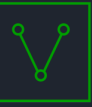


# Pivoting



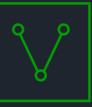
# Storm Operations

Operation	Meaning	Common Storm Operator	UI Equivalent
Lift	Select data (nodes) from Synapse	Query bar - Storm	Query bar - Lookup / Text Search query and copy menu options
Pivot	Move between nodes that share the same <b>property value</b>	-> or <- *	Explore button pivot menu option
Traverse	Move between nodes that are linked by an <b>edge</b>	-(*)> or <(*)-	Explore button
Filter	Include / exclude a subset of nodes	+ or -	n/a (column filters; query / select menu options)
Run	Execute a Storm command	<command>	Node Action
Modify / Edit	Modify or delete properties Add or remove tags Add nodes	[ ] or [ ( ) ]	Inline property edit; delete menu option Add / remove tags menu options Lookup or Auto Add / Add Node



# Pivoting

- Primary way to navigate data in Synapse
- Uses intuitive "arrow" symbol ( -> )
- Move between nodes that share a **property value**
  - o If properties are also the same **type**...
  - o ...Synapse **automatically** identifies these relationships
- A pivot represents an **implicit** connection / relationship between nodes
  - o Don't need to create these connections - they just exist



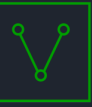
# Pivoting - Type Awareness

- Nodes with **properties** with the same **type** have an **implicit relationship**

Form / Property	Type
<b>inet:fqdn</b>	inet:fqdn
:domain	inet:fqdn
:host	str
:issuffix	bool
:iszone	bool
:zone	inet:fqdn

Form / Property	Type
<b>inet:dns:a</b>	inet:dns:a
:fqdn	inet:fqdn
:ipv4	inet:ipv4

Form / Property	Type
<b>inet:ipv4</b>	inet:ipv4
:asn	inet:asn
:dns:rev	inet:fqdn
:latlong	geo:latlong
:loc	loc
:place	geo:place
:type	str



# Pivoting - Type Awareness

- Nodes with **properties** with the same **type** and **value** are **connected**

Form / Property	Value
<b>inet:fqdn</b>	work.viewdns.ml
:domain	viewdns.ml
:host	work
:issuffix	False
:iszone	False
:zone	viewdns.ml

Form / Property	Value
<b>inet:dns:a</b>	inet:dns:a
:fqdn	work.viewdns.ml
:ipv4	195.20.50.249

Form / Property	Value
<b>inet:ipv4</b>	195.20.50.249
:asn	31624
:latlong	52.3824,4.8995
:loc	nl
:type	unicast

**Protip:** Type awareness is how "Explore" in Synapse (or pivoting in Storm) works!



# Data Model - Type Awareness

- "Properties" and "Referenced By" show all type-based interconnections
- Use to see:
  - o What is this form connected to?
  - o How is it connected?
  - o How can I navigate between forms?

**Tip:** you can also use the Explore button to see "what's connected".

FQDN: `inet:fqdn` [Lift in Research Tool](#) [docs link](#)

A Fully Qualified Domain Name (FQDN). type: `inet:fqdn`  
example: `vertex.link`

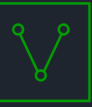
Properties

name	ro	type	doc
:domain	—	<code>inet:fqdn</code>	The parent domain for the FQDN.
:host	—	<code>str</code>	The host part of the FQDN.
:issuffix		<code>bool</code>	True if the FQDN is considered a suffix.
:iszone		<code>bool</code>	True if the FQDN is considered a zone.
:zone		<code>inet:fqdn</code>	The zone level parent for this FQDN.
.created	—	<code>time</code>	The time the node was created in the cortex.
.seen		<code>ival</code>	The time interval for first/last observation of the node.

Referenced By

form	prop	doc
<code>biz:deal</code>	:buyer:orgfqdn	The reported <code>inet:fqdn</code> of the buyer org.
<code>biz:deal</code>	:seller:orgfqdn	The reported <code>inet:fqdn</code> of the seller org.
<code>biz:product</code>	:madeby:orgfqdn	The reported <code>inet:fqdn</code> of the product manufacturer.
<code>biz:stake</code>	:orgfqdn	The org FQDN as reported by the source of the vitals.
<code>crypto:x509:cert</code>	:identities:fqdns	The fused list of FQDNs identified by the cert CN and SANs.
<code>inet:dns:a</code>	:fqdn	The domain queried for its DNS A record.
<code>inet:dns:aaaa</code>	:fqdn	The domain queried for its DNS AAAA record.
<code>inet:dns:cname</code>	:fqdn	The domain queried for its CNAME record.
<code>inet:dns:cname</code>	:cname	The domain returned in the CNAME record.
<code>inet:dns:mx</code>	:fqdn	The domain queried for its MX record.
<code>inet:dns:mx</code>	:mx	The domain returned in the MX record.
<code>inet:dns:ns</code>	:zone	The domain queried for its DNS NS record.
<code>inet:dns:ns</code>	:ns	The domain returned in the NS record.
<code>inet:dns:query</code>	:name:fqdn	A Fully Qualified Domain Name (FQDN).
<code>inet:dns:request</code>	:query:name:fqdn	A Fully Qualified Domain Name (FQDN).
<code>inet:dns:rev</code>	:fqdn	The domain returned in the PTR record.
<code>inet:dns:rev6</code>	:fqdn	The domain returned in the PTR record.





# Pivoting in Storm

- A pivot requires:
  - o The **source** (nodes or properties you're coming from)
  - o A pivot **operator** ( `->` is the most common)
  - o The **target** of the pivot (nodes or properties you're pivoting to)
- Pivoting navigates **away** from the source **to** the target

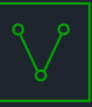
```
inet:fqdn=vertex.link -> inet:dns:a:fqdn
```

Source:

- the FQDN 'vertex.link'

Target:

- `inet:dns:a` nodes with  
matching `:fqdn` property



# Source and Target

```
inet:dns:a:fqdn=vertex.link :ipv4 -> inet:ipv4
```

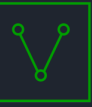
- **Source:** `:ipv4` property (**relative** property name) of `inet:dns:a` nodes
- **Target:** any matching `inet:ipv4` nodes

```
inet:fqdn=hugesoft.org :zone -> inet:fqdn:zone
```

- **Source:** `:zone` property of FQDN `hugesoft.org`
- **Target:** any `inet:fqdn` with the same `:zone` property value

```
file:bytes#rep.mandiant.ap1 -> *
```

- **Source:** `file:bytes` nodes tagged `#rep.mandiant.ap1`
- **Target:** wildcard - **any** node matching **any** property of **any** source `file:bytes`



# Explicit vs. Implicit Syntax

- **Explicit:** specifies the source and target forms and **properties**
  - Tell Synapse the **exact** navigation you want to perform

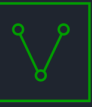
```
inet:fqdn=vertex.link -> inet:dns:request:query:name:fqdn :exe -> file:bytes
```

- **Implicit:** specify source and target **forms only**
  - Synapse figures out which properties you mean (type awareness)

```
inet:fqdn=vertex.link -> inet:dns:request -> file:bytes
```

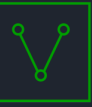
- When using a **wildcard** explicit vs. implicit does not apply

```
file:bytes#rep.mandiant.apt1 -> *
```

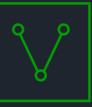


# Pivot Examples

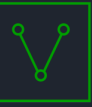
Kind of Pivot	Example	Question
Wildcard pivot out	<code>inet:ipv4=1.2.3.4 -&gt; *</code>	Show me everything <b>this node points to</b>
Wildcard pivot in	<code>inet:ipv4=1.2.3.4 &lt;- *</code>	Show me everything <b>that points to this node</b>
Pivot out	<code>inet:fqdn=woot.com -&gt; inet:dns:a:fqdn</code>	Show me the DNS A records for this FQDN
Pivot out	<code>inet:fqdn=woot.com -&gt; inet:dns:a*</code>	Show me the DNS A and AAAA records for this FQDN
Pivot out	<code>inet:fqdn=woot.com -&gt; inet:fqdn:zone</code>	Show me the FQDN records where this FQDN is the zone
Pivot out	<code>inet:fqdn=woot.com -&gt; (inet:dns:a, inet:dns:ns)</code>	Show me the DNS A and NS records for this FQDN
Pivot out	<code>file:bytes:md5=6de25e21cfda939dda1a41a326f5de10 -&gt; it:host:activity</code>	Show me all the execution activity associated with this file



# Pivoting - Demo



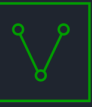
# Tag Pivots



# Tag Pivots

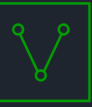
Kind of Pivot	Example	Question
Pivot from tags (from <code>syn:tag</code> nodes to tagged nodes)	<code>syn:tag=cno.ma1 -&gt; *</code>	Show me the nodes that have this tag
Pivot to tags (from tagged nodes to <code>syn:tag</code> nodes)	<code>inet:ipv4=1.2.3.4 -&gt; #</code>	Show me the <b>leaf</b> <code>syn:tag</code> nodes for the tags on this node
Pivot to tags (from tagged nodes to <code>syn:tag</code> nodes)	<code>inet:ipv4=1.2.3.4 -&gt; #*</code>	Show me <b>all</b> the <code>syn:tag</code> nodes for the tags on this node

**Tip:** Because pivoting to/from `syn:tag` nodes uses special behavior by default, you must use explicit syntax when pivoting to/from tag **properties**.

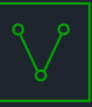


# Tag Pivots - Demo



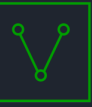


# Traversal



# Storm Operations

Operation	Meaning	Common Storm Operator	UI Equivalent
Lift	Select data (nodes) from Synapse	Query bar - Storm	Query bar - Lookup / Text Search query and copy menu options
Pivot	Move between nodes that share the same <b>property value</b>	-> or <- *	Explore button pivot menu option
Traverse	Move between nodes that are linked by an <b>edge</b>	-(*)> or <(*)-	Explore button
Filter	Include / exclude a subset of nodes	+ or -	n/a (column filters; query / select menu options)
Run	Execute a Storm command	<command>	Node Action
Modify / Edit	Modify or delete properties Add or remove tags Add nodes	[ ] or [ ( ) ]	Inline property edit; delete menu option Add / remove tags menu options Lookup or Auto Add / Add Node



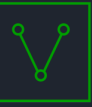
# Edge Traversal

- Lightweight (light) edges connect nodes that do not have properties in common
  - Often a generic relationship ("references", "seen by")
- Traversal navigates ("walks") between nodes joined by a light edge
- A light edge is an **explicit** connection / relationship between nodes
  - Must explicitly create or remove edges between nodes



# Traversal in Storm

- Edge traversal in Storm requires:
  - o The **source** (nodes you're coming from)
  - o The traversal "**arrow**" operator: `-( )>` or `<( )-`
    - Correct "**direction**" for the edge relationship
    - Edge **name/names** (or wildcard `*`)
  - o The **target** (nodes you're traversing to)
- Traversal navigates **from** the source **to** the target



# Data Model - Edges

- View in Data Model Explorer
- Recommended edges / edge use

DATA MODEL EXPLORER DOCUMENTATION POWER-UPS TAG EXPLORER VERSION CHANGELOG

Search

- (about)>
- (acquired)>
- (addresses)>
- (contains)>
- (detects)>
- (follows)>
- (found)>
- (has)>**
- (includes)>
- (ipwhois)>
- (leaked)>

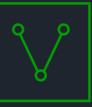
- Show Forms
- Show Types
- Show Interfaces
- Show Edges
- Show deprecated

-(has)>

Show edge model elements.

### Edge Usages

source	verb	target
belief:system	-(has)>	belief:tenet
econ:bank:statement	-(has)>	econ:acct:payment
meta:ruleset	-(has)>	meta:rule
ou:org	-(has)>	*
ps:contact	-(has)>	*
ps:person	-(has)>	*
sci:evidence	-(has)>	*
sci:observation	-(has)>	*



# Source and Target

```
media:news:publisher:name=eset -(refs)> *
```

- **Source:** All `media:news` nodes whose `:publisher:name` property is 'eset'
- **Edge:** `refs` ("references"); direction is "forward"
- **Target:** wildcard - **any** nodes linked to the article(s) by a `refs` edge

```
media:news:publisher:name=eset -(refs)> inet:fqdn
```

- **Source:** All `media:news` nodes whose `:publisher:name` property is 'eset'
- **Edge:** `refs` ("references"); direction is "forward"
- **Target:** any `inet:fqdn` nodes linked to the article(s) by a `refs` edge

```
inet:fqdn=music.todayusa.org <(*)- meta:source
```

- **Source:** the FQDN `music.todayusa.org`
- **Edge:** wildcard - **any** edge; direction is "backward"
- **Target:** any data source (`meta:source` node) linked to the FQDN by **any** edge



# Common Edges - refs

- refs ("references") use cases:

- o Link articles to things "referenced" by an article:

```
media:news=<guid> -(refs)> *
```

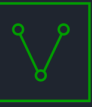
- o Link a node with a text property to things scraped from the text:

```
inet:service:message=<guid> -(refs)> *
```

- o Find all the things that "reference" an MD5 hash:

```
hash:md5=<hash> <(refs)- *
```

```
hash:md5=<hash> <(refs)- media:news
```



# Common Edges - seen

- seen use cases:

- o Link a data source to all nodes observed by the source:

```
meta:source=<guid> -(seen) *
```

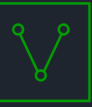
- o Find all the sources that "saw" an indicator

```
inet:dns:a=(woot.com,1.2.3.4) <(seen) - *
```

```
inet:dns:a=(woot.com,1.2.3.4) <(seen) - meta:source
```

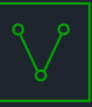
**Protip:** Light edges are documented in **Data Model Explorer**. Light edges created by Power-Ups are documented in the Power-Up Help.



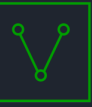


# Traversal Examples

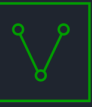
Example	Question
<code>inet:fqdn=vertex.link &lt;(*)- *</code>	Show me every node connected to this FQDN by any edge
<code>inet:fqdn=vertex.link &lt;(refs)- *</code>	Show me every node that references this FQDN
<code>inet:fqdn=vertex.link &lt;(refs)- media:news</code>	Show me every article that references this FQDN
<code>inet:fqdn=vertex.link &lt;( (refs, seen) )- *</code>	Show me every node that references or has seen this FQDN
<code>media:news=&lt;guid&gt; -(refs)&gt; (inet:fqdn, inet:ipv4)</code>	Show me the FQDNs and IPv4s referenced by this article
<code>media:news=&lt;guid&gt; -(refs)&gt; hash:*</code>	Show me all the hashes referenced by this article



# Traversal - Demo



# Additional Operations



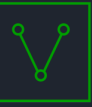
# Pivot and Traverse

- Pivots and traversals navigate different kinds of connections
  - o Property-based vs edge-based
- Use a "double arrow" to perform both at once
  - o Must use wildcard as a target

```
inet:ipv4=1.2.3.4 --> *
```

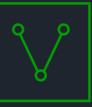
```
inet:fqdn=vertex.link <-- *
```

**Tip:** Using the wildcard as a target is an easy way to use Storm to explore / see what is connected to your source node(s).



# Pivot / Traverse and Explore

Operation	Meaning	Common Storm Operator	UI Equivalent
Pivot	Move between nodes that share the same <b>property value</b>	-> or <- *	<b>Explore button</b>
Traverse	Move between nodes that are linked by an <b>edge</b>	-(*)> or <(*)-	<b>Explore button</b>



# Pivot / Traverse and Explore

Operation	Meaning	Common Storm Operator	UI Equivalent
Pivot	Move between nodes that share the same <b>property value</b>	-> or <- *	<b>Explore button</b> pivot menu option
Traverse	Move between nodes that are linked by an <b>edge</b>	-(*)> or <(*)-	<b>Explore button</b>

– The **Explore** button does all of the above at once:

- Wildcard pivot out ( -> \* )
  - Wildcard traverse (walk) out ( -(\*)> )
  - Wildcard pivot in ( <- \* )
  - Wildcard traverse (walk) in ( <(\*)- )
- Pivot out and walk ( --> \* )
- Pivot in and walk ( <-- \* )



Explore button



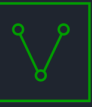
# Join Operations

- Pivot and traversal both navigate away from the source to the target
  - o The nodes you see (your "working set") changes
- Sometimes you want to see both
- Pivot / traversal operators can use an embedded plus sign ( + ) to keep the source nodes

```
inet:dns:a:fqdn=vertex.link -+> ( inet:fqdn, inet:ipv4 )
```

```
inet:ipv4=1.2.3.4 <+(seen)- meta:source
```

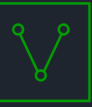
```
file:bytes=<sha256> <+-- *
```



# Synapse UI and Storm

Synapse UI (Explore)	Storm Pivot / Traverse
Good for Exploring - "I don't know what's connected"	Good for targeted navigation - "I know exactly where I want to go"
No typing!	Typing...but ways to simplify
Supports standard pivots - between primary / secondary properties	Supports <b>all</b> pivot formats using explicit syntax
May need to navigate / display large numbers of nodes to get to where you want to go	Only navigate / display exactly what you need





# Summary

- In Synapse, **pivoting** navigates between nodes that share a **property value**
  - o Storm uses the "arrow" symbol for pivots ( -> )
- **Explicit syntax** tells Synapse **exactly** how to pivot
  - o Source / target **forms** and **properties**
- Many pivots can use **implicit syntax**
  - o Source / target **forms** only
- **Traversal** navigates between nodes connected by a **light edge**
  - o Storm uses an arrow with an edge name (or wildcard) for traversal ( -(\*)> )
- The Synapse **Explore** button uses pivots and traversals to automatically navigate for you