
jaga
CLIMATE
DESIGNERS

#OurGiantsAreFemale

Margaret Ingels

- October 25, 1892 – December 13, 1971
- First female engineering graduate from the University of Kentucky

- Joined the American Society of Heating and Ventilating Engineers Research Lab
- Developed the “effective temperature” scale to incorporate humidity and air movement in the equation for human comfort

- “Petticoats and Slide Rules” is a famous speech and inspiration to women in engineering

A Tree Full of Object Data

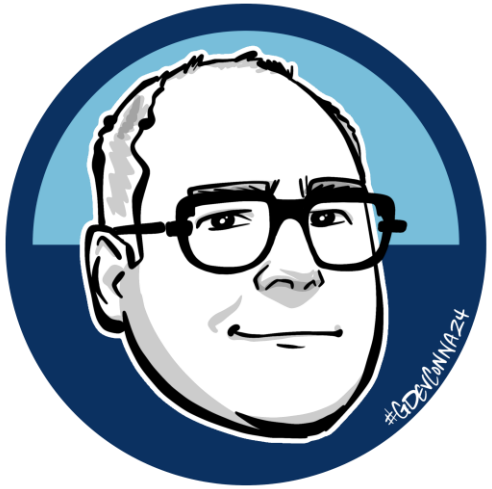
Decomposing object data into a tree structure using JSON and SQLite

Agenda

- About Me
- Warning
- Problem Statement
- Solution
  - Object to JSON
  - JSON to tree
- Use Cases
- Bonus
- Links

About Me

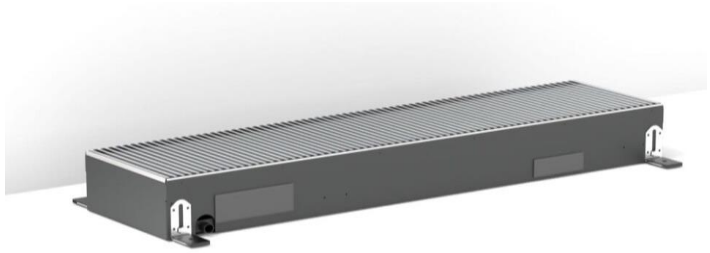
- **Stefan Lemmens**
  - Test development engineer at Jaga Climate Designers
  - LabVIEW Developer since 1999 (LabVIEW 5.0)
  - LabVIEW Champion since 2021
  - Member of SAS LabVIEW Mastermind
  - Created a leaderboard on Advent of Code and Discord server for LabVIEW developers
  - LabVIEW shortcuts booklet
  - Mostly use Actor Framework (basics to PPL tutorial)
  - Mail : stefanlemmens.be@gmail.com
  - NI Forum : StefanLemmens
  - Twitter : @stefan_Lemmens

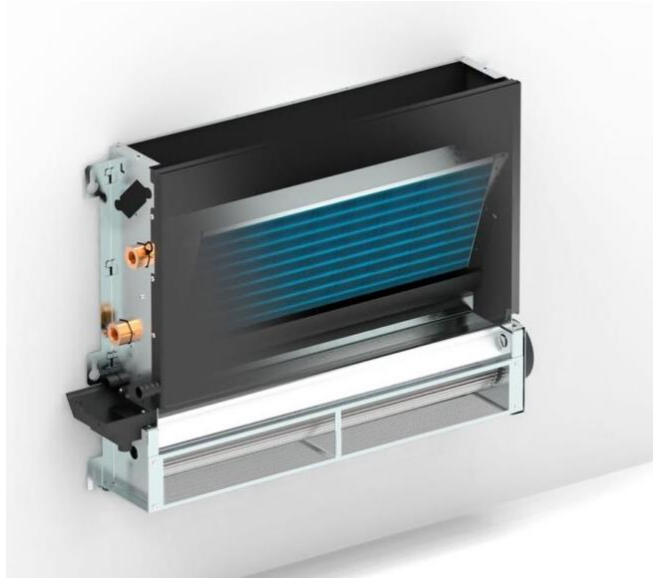


jaga CLIMATE DESIGNERS

Ensuring a better indoor climate without damaging the outdoor climate.
That's our main goal as Jaga Climate Designers!

Jaga Products





jaga CLIMATE DESIGNERS

Warning

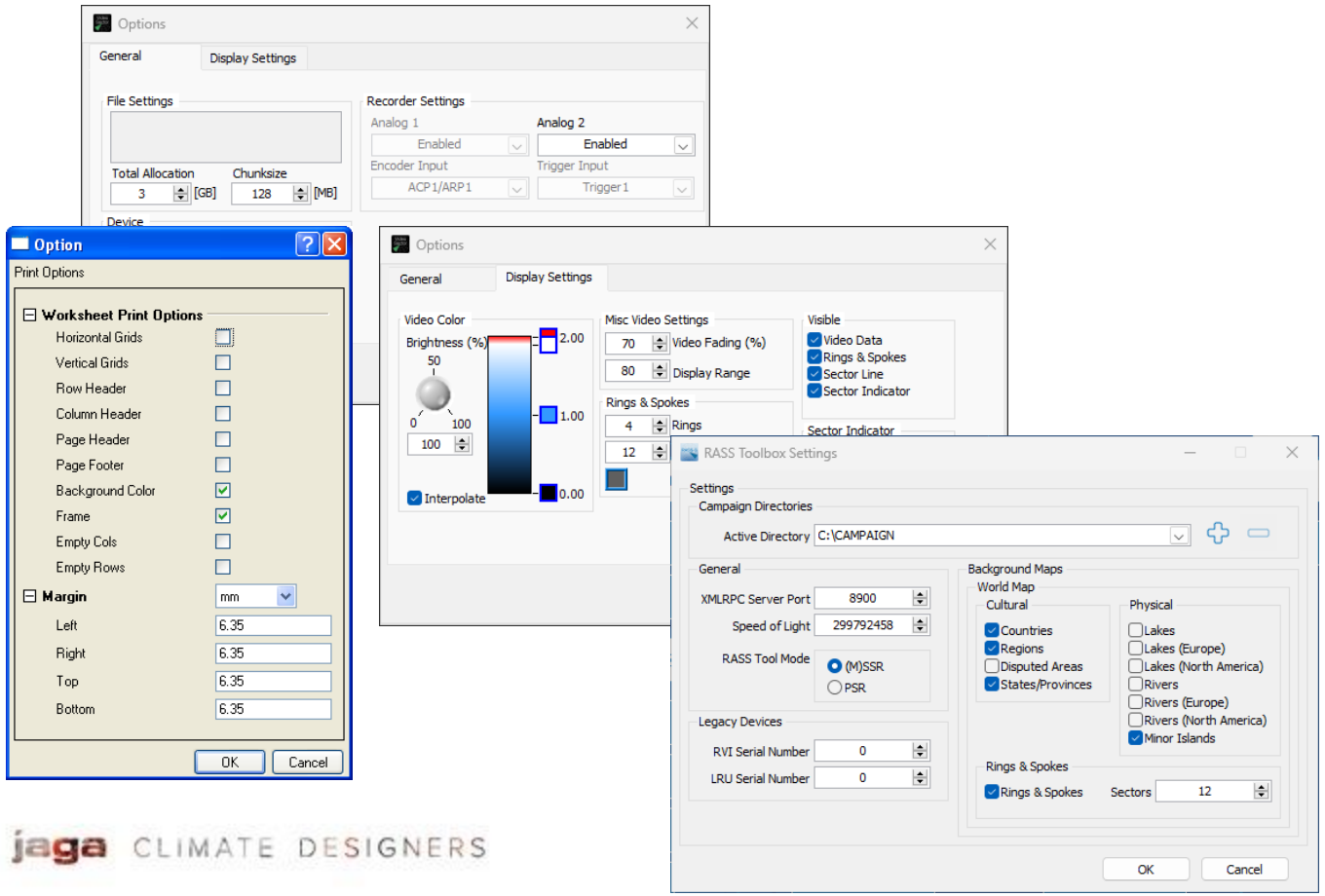
- This works for me
- Still room for improvements
- This will break encapsulation
  - ➢ Use it carefully

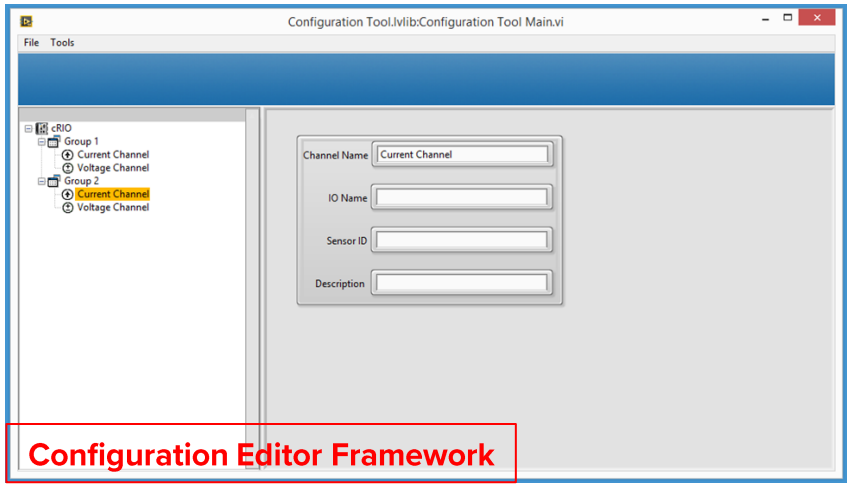
Warning

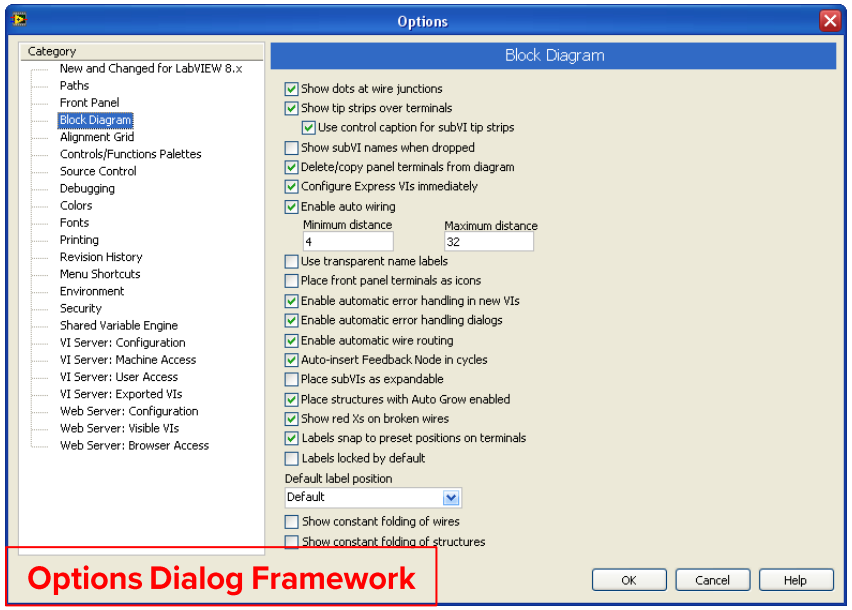
- This works for me
- Still room for improvements
- This will break encapsulation
  - ➢ Use it carefully

Problem Statement

- Different applications/test have different settings
- Configuration clusters/objects can contain a lot of parameters
- A different UI is needed each time







jaga CLIMATE DESIGNERS

Requirements

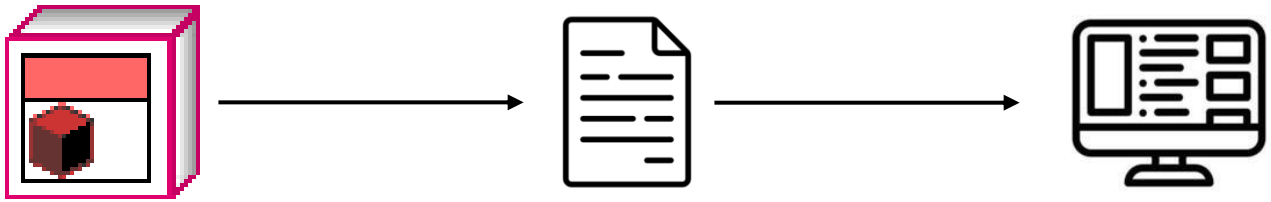
- 1 **Generic** UI

- User-friendly

- Take a (configuration) object and
  - ✓ Display
  - ✓ Edit
  - ✓ Read from file
  - ✓ Save to file

Solution

- Part 1 : Object to/from text/file
- Part 2 : Display & edit the data

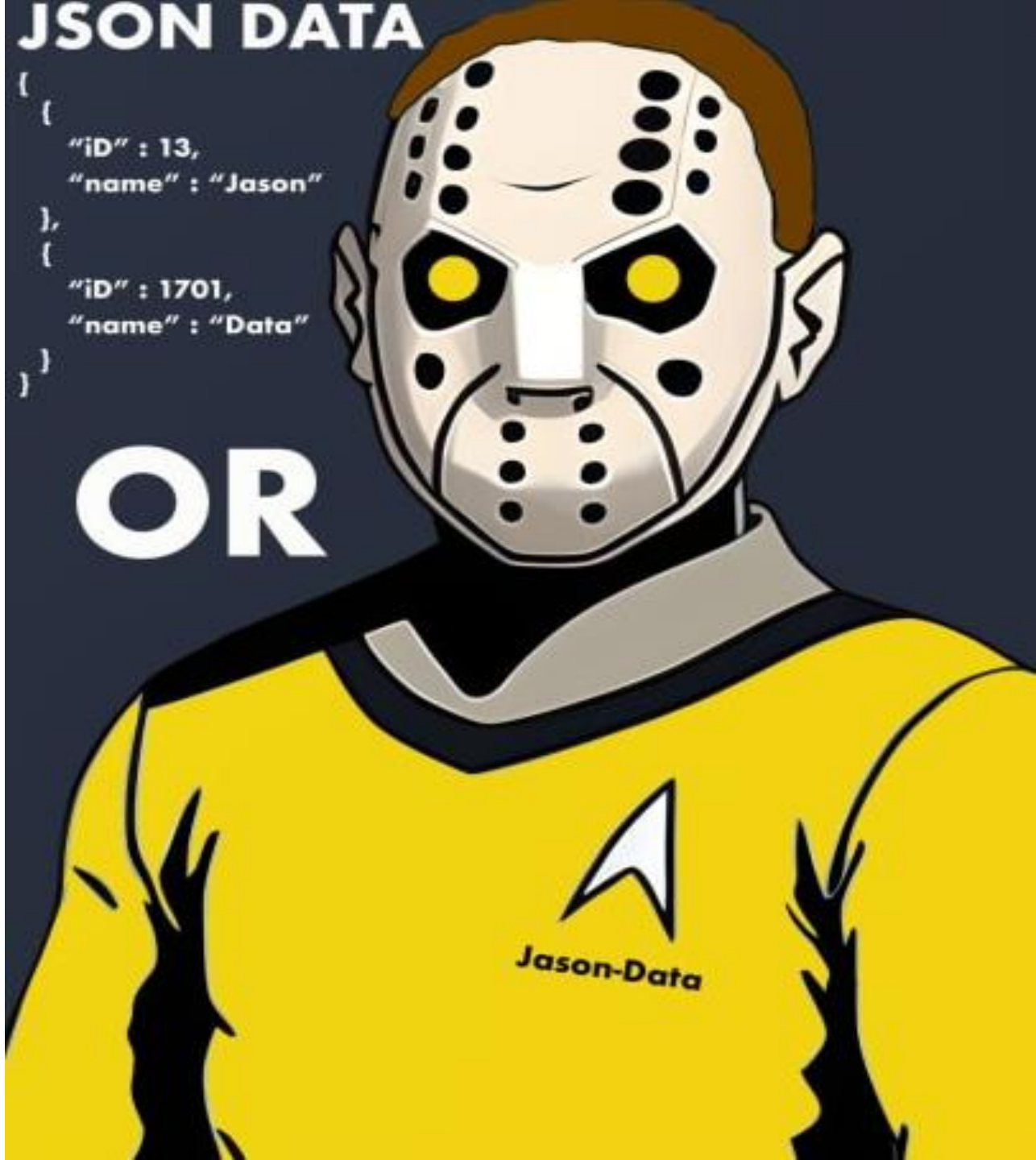


File Format

- I wanted to switch from XML to JSON

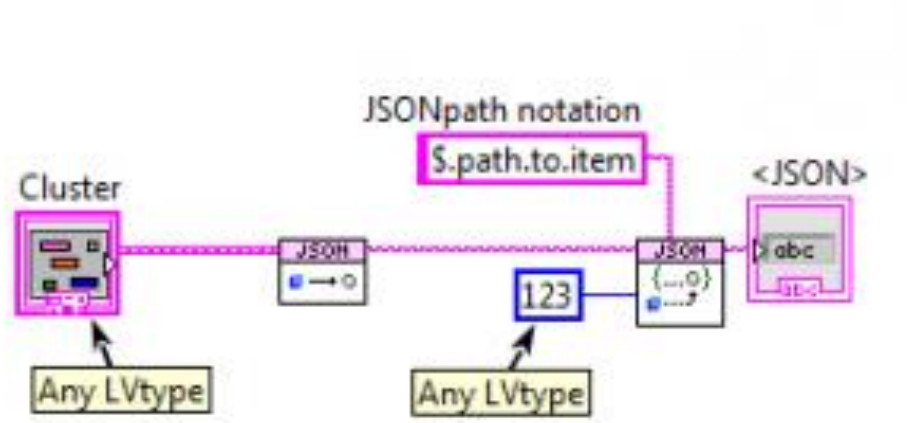
- Why????

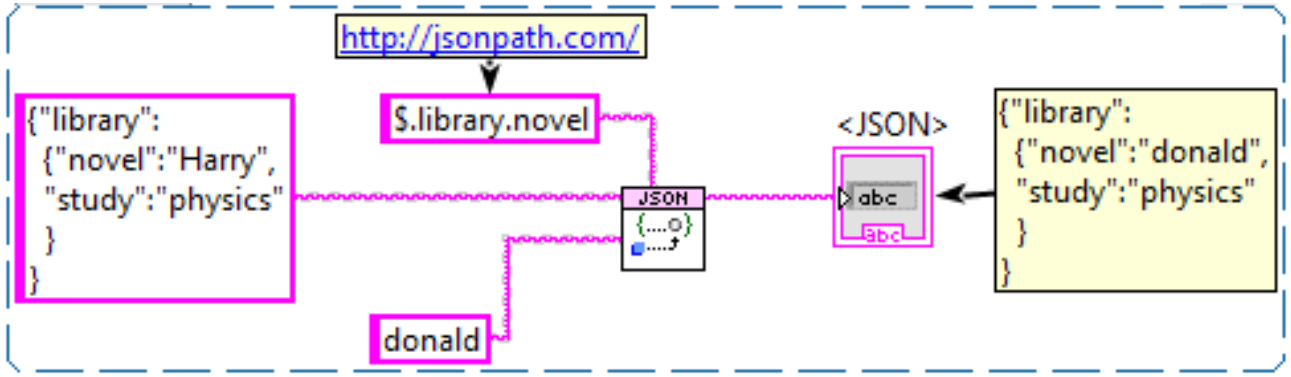
  - ✓ Easier to read and parse
  - ✓ Better support for (empty) arrays
  - ✓ Supports optional elements easier
  - ✓ Way to represent objects
  - ✓ Better suited for data storage
  - ✓ Used by Jaga Device Controllers



JSONtext

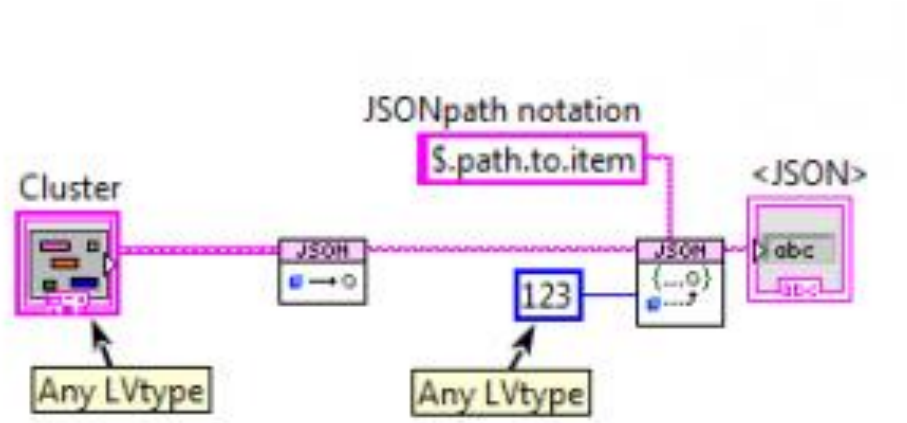
- JSONtext by JDP
  - ✓ Uses JSON Path notation to access (sub)items
  - ✓ Very good for everything with JSON
  - ✓ Now also support JSON scheme validation

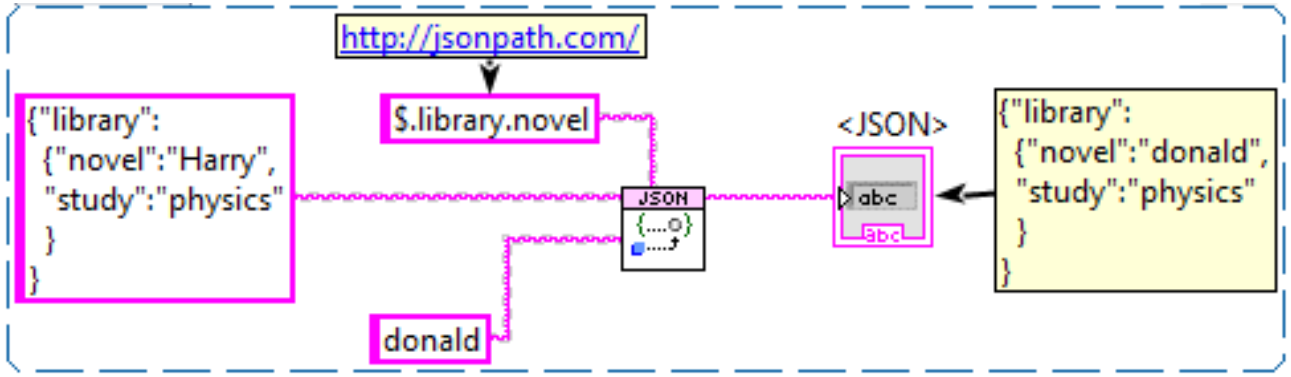




JSONtext

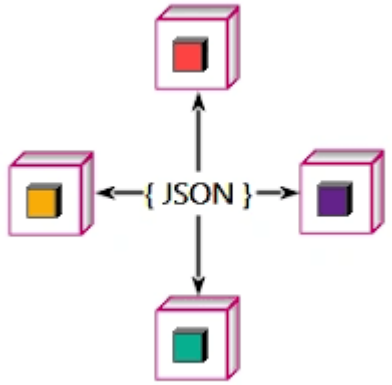
- JSONtext by JDP
  - ✓ Uses JSON Path notation to access (sub)items
  - ✓ Very good for everything with JSON but doesn't support objects
  - ✓ Now also support JSON scheme validation

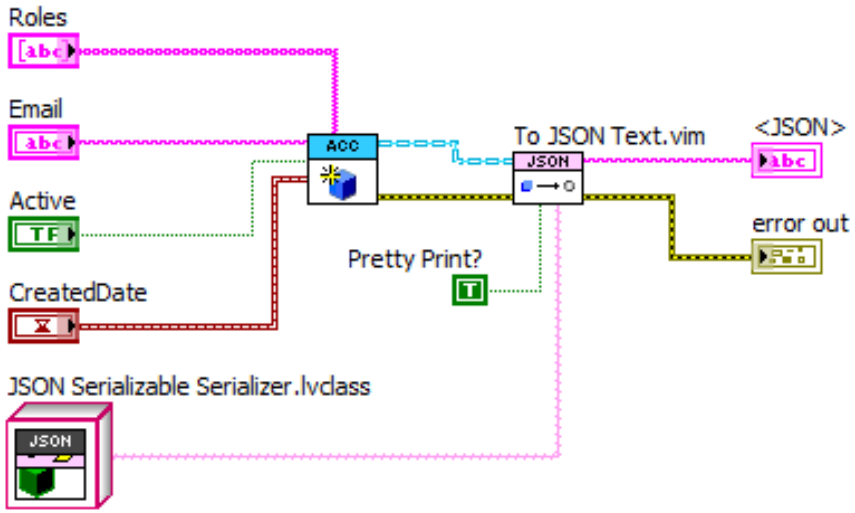


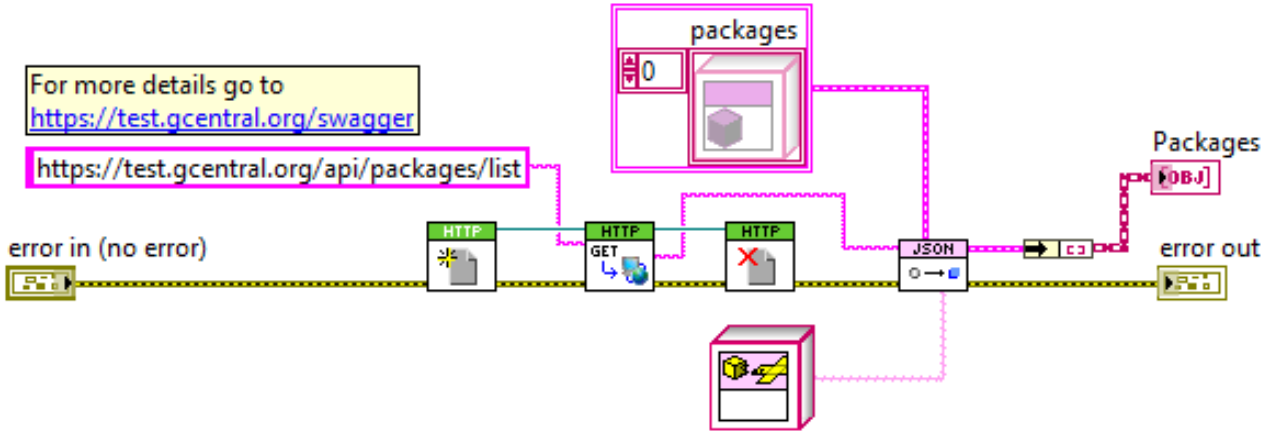


JSONtext Object Serialization

- VIPM toolkit by PNR
- Extension for JSONtext to (de)serialize objects
- Easy to use
- Breaks encapsulation !!!!





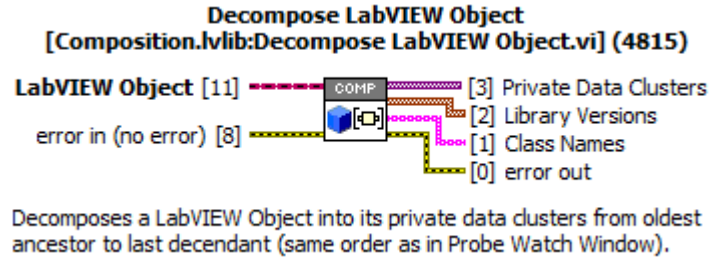


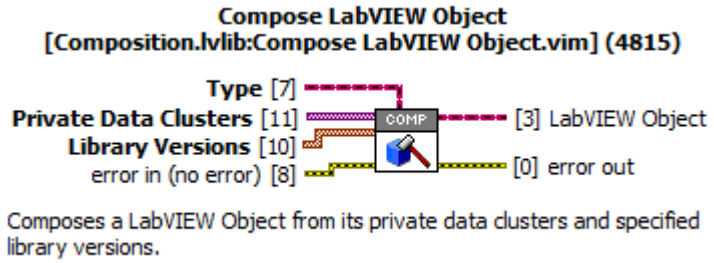
LabVIEW Composition

- Another VIPM toolkit by PNR
- Provides functions to compose and decompose
  - objects
  - clusters
  - maps and sets

- Breaks encapsulation
- Makes private data accessible
- Easy to use

- Used by JSONText object serialization



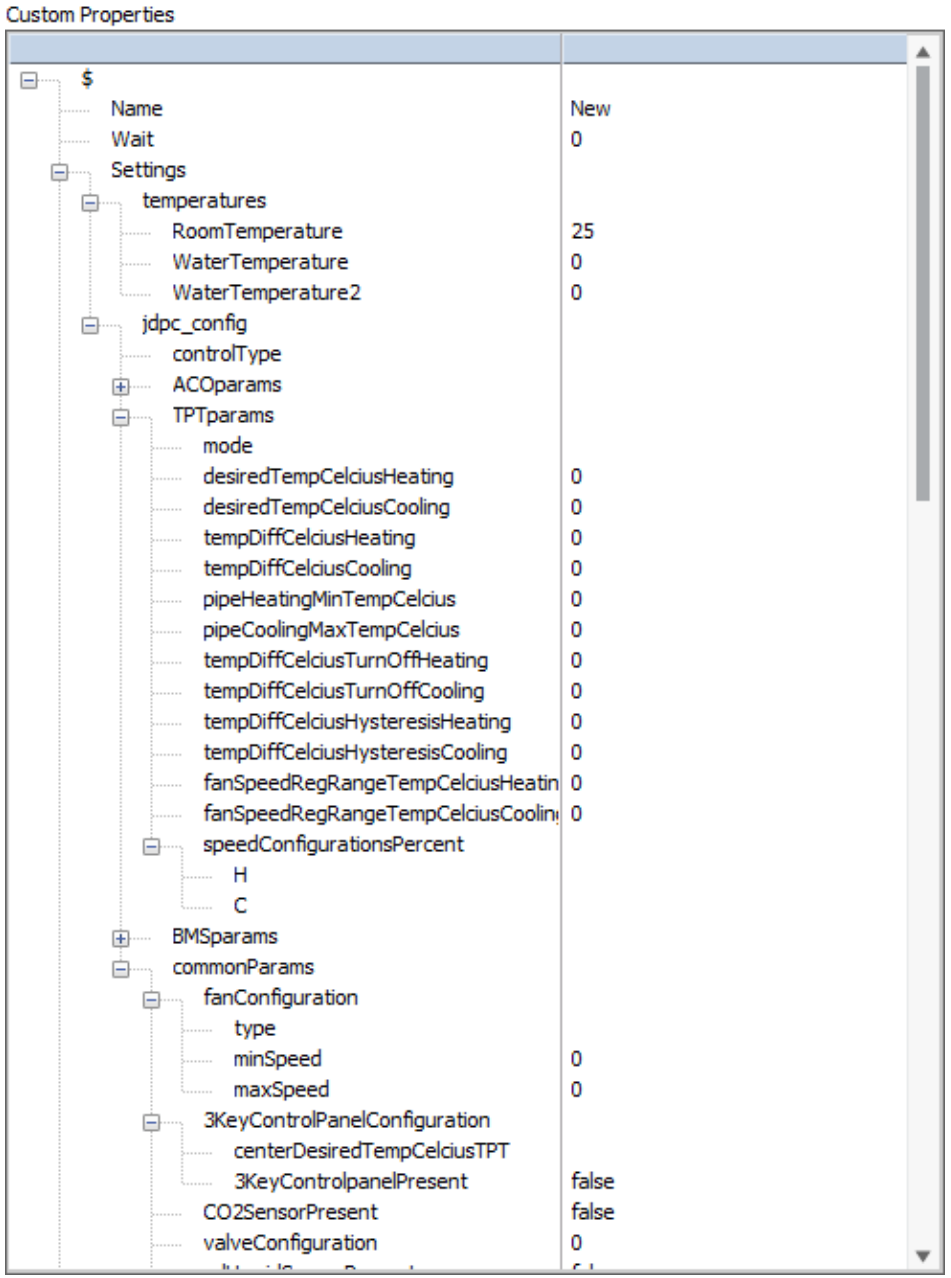


Displaying the data

- Generic way to display all kinds of data in a user-friendly way

Tree Structure

- Can handle multiple levels
- Can handle different data types
- Commonly used to display JSON



Show JSON in a tree

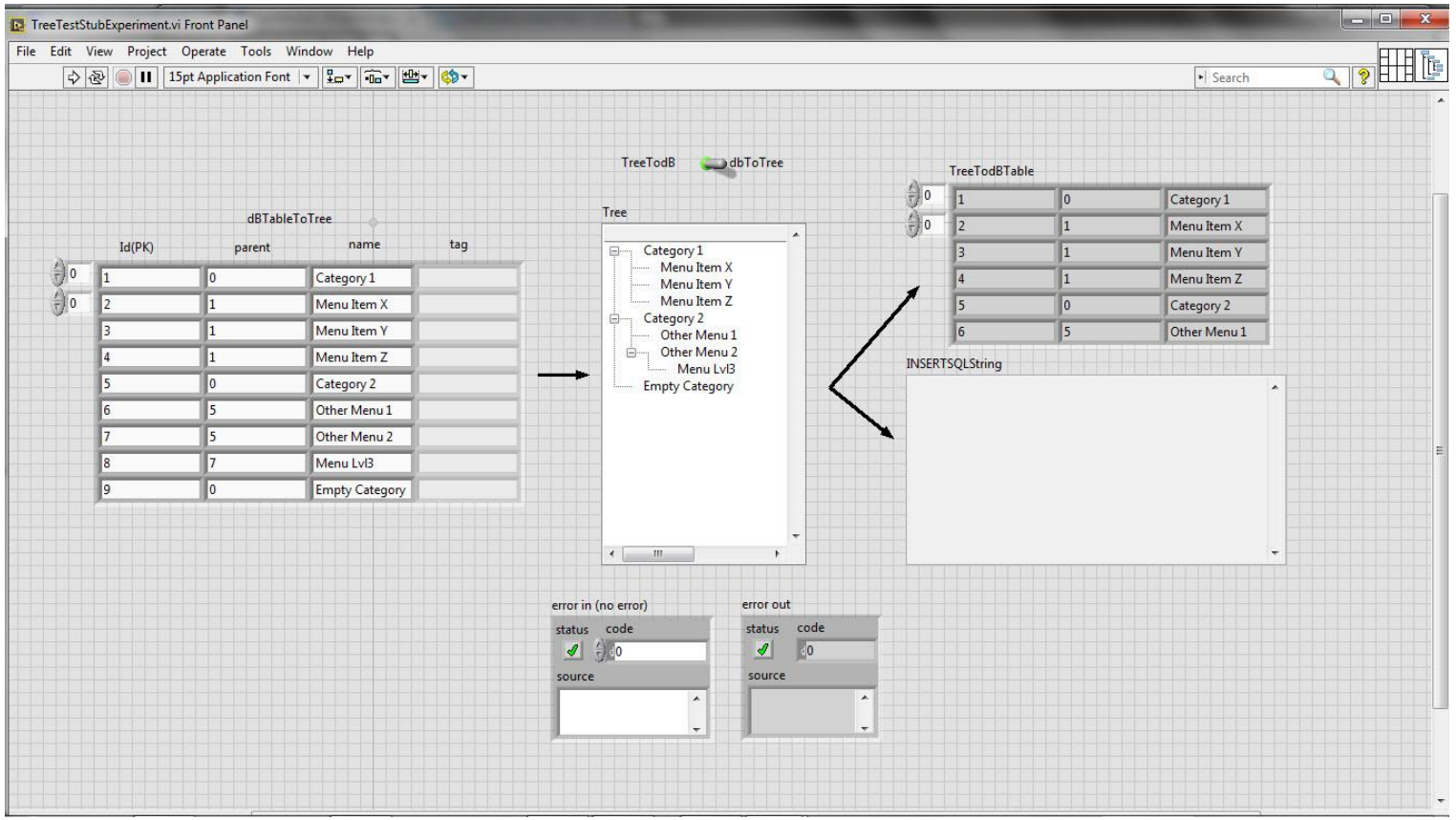
- Let’s ask ChatGPT

Show JSON in a tree

- Let’s ask ChatGPT

Tree Control - A little present

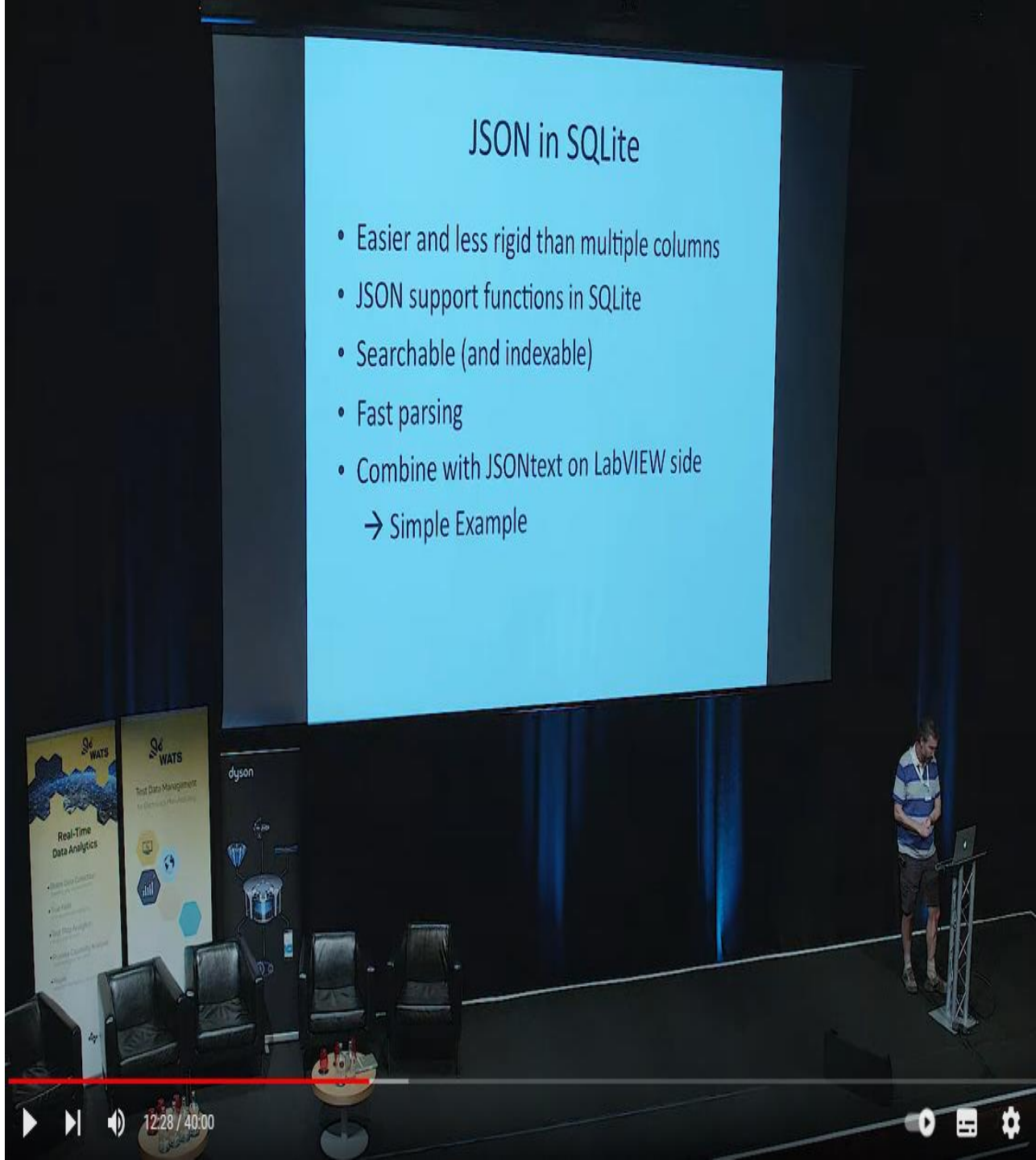
- From Steve Watts (Random Ramblings on LabVIEW Design blog)
- Create a tree hierarchy from a database table



JSON to database table

- Combine JSONtext and SQLite to create the database table

- Presented at GDevCon#2
  by Dr. James Powell

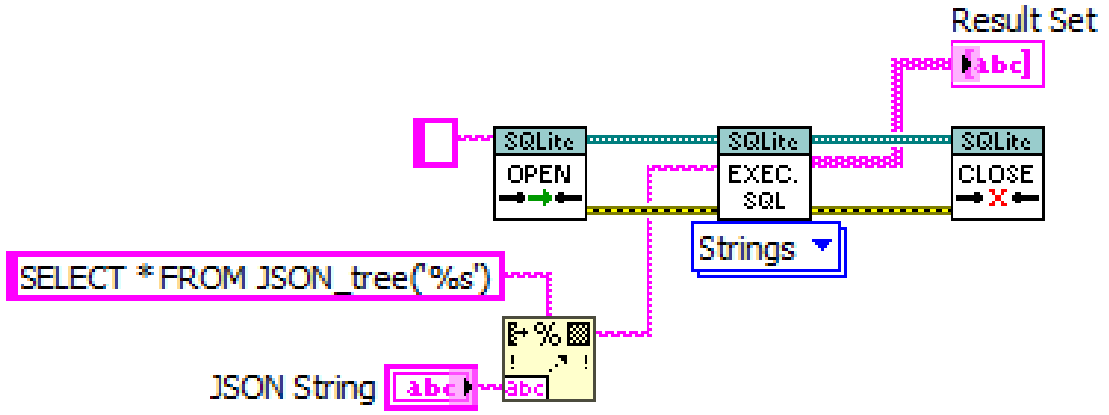


Example data


```
{
  "Actors": [{
      "name": "Tom Cruise",
      "age": 56,
      "Born At": "Syracuse, NY",
      "Birthdate": "July 3, 1962",
      "photo": "https://jsonformatter.org/img/tom-cruise.jpg",
      "wife": null,
      "weight": 67.5,
      "hasChildren": true,
      "hasGreyHair": false,
      "children": [
        "Suri",
        "Isabella Jane",
        "Connor"
      ]
    },
    {
      "name": "Robert Downey Jr.",
      "age": 53,
      "Born At": "New York City, NY",
      "Birthdate": "April 4, 1965",
      "photo": "https://jsonformatter.org/img/Robert-Downey-Jr.jpg",
      "wife": "Susan Downey",
      "weight": 77.1,
      "hasChildren": true,
      "hasGreyHair": false,
      "children": [
        "Indio Falconer",
        "Avri Roel",
        "Exton Elias"
      ]
    }
  ]
}
```

JSON_tree

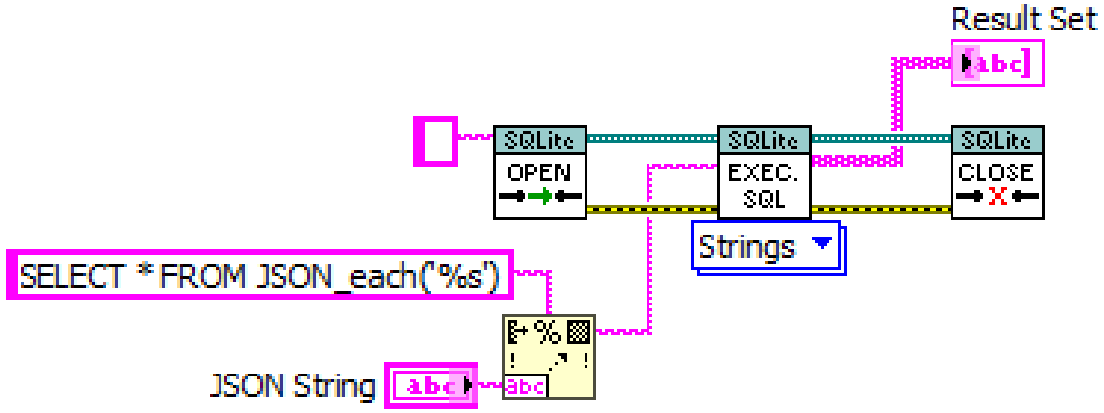
- Can query JSON data
- Retuns table with one row for each element
- Walks recursively through JSON substructures



	key	value	type	atom	id	parent	fullkey	path
1	*NULL*	{"Actors":[{"name":"Tom Cruise","age":56,"Born...	object	*NULL*	0	*NULL*	$	$
2	Actors	[{"name":"Tom Cruise","age":56,"Born ...	array	*NULL*	2	0	$.Actors	$
3	0	{"name":"Tom Cruise","age":56,"Born ...	object	*NULL*	3	2	$.Actors[0]	$.Actors
4	name	Tom Cruise	text	Tom Cruise	5	3	$.Actors[0].name	$.Actors[0]
5	age	56	integer	56	7	3	$.Actors[0].age	$.Actors[0]
6	Born At	Syracuse, NY	text	Syracuse, NY	9	3	$.Actors[0].Born At	$.Actors[0]
7	Birthdate	July 3, 1962	text	July 3, 1962	11	3	$.Actors[0].Birthdate	$.Actors[0]
8	photo	https://jsonformatter.org/img/tom-cruise.jpg	text	https://jsonformatter.org/img/tom-cruise.jpg	13	3	$.Actors[0].photo	$.Actors[0]
9	wife	*NULL*	null	*NULL*	15	3	$.Actors[0].wife	$.Actors[0]
10	weight	67.5	real	67.5	17	3	$.Actors[0].weight	$.Actors[0]
11	hasChildren	1	true	1	19	3	$.Actors[0].hasChildren	$.Actors[0]
12	hasGreyHair	0	false	0	21	3	$.Actors[0].hasGreyHair	$.Actors[0]
13	children	["Suri","Isabella Jane","Connor"]	array	*NULL*	23	3	$.Actors[0].children	$.Actors[0]
14	0	Suri	text	Suri	24	23	$.Actors[0].children[0]	$.Actors[0].children

JSON_each

- Can query JSON data
- Retuns table with one row for each element
- Only walks one level



	key	value	type	atom	id	parent	fullkey	path
1	name	Tom Cruise	text	Tom Cruise	2	NULL	$.name	$
2	age	56	integer	56	4	NULL	$.age	$
3	Born At	Syracuse, NY	text	Syracuse, NY	6	NULL	$.Born At	$
4	Birthdate	July 3, 1962	text	July 3, 1962	8	NULL	$.Birthdate	$
5	photo	https://jsonformatter.org/img/tom-cruise.jpg	text	https://jsonformatter.org/img/tom-cruise.jpg	10	NULL	$.photo	$
6	wife	NULL	null	NULL	12	NULL	$.wife	$
7	weight	67.5	real	67.5	14	NULL	$.weight	$
8	hasChildren	1	true	1	16	NULL	$.hasChildren	$
9	hasGreyHair	0	false	0	18	NULL	$.hasGreyHair	$
10	children	["Suri","Isabella Jane","Connor"]	array	NULL	20	NULL	$.children	$

JSON_extract

- Extracts and returns one or more values

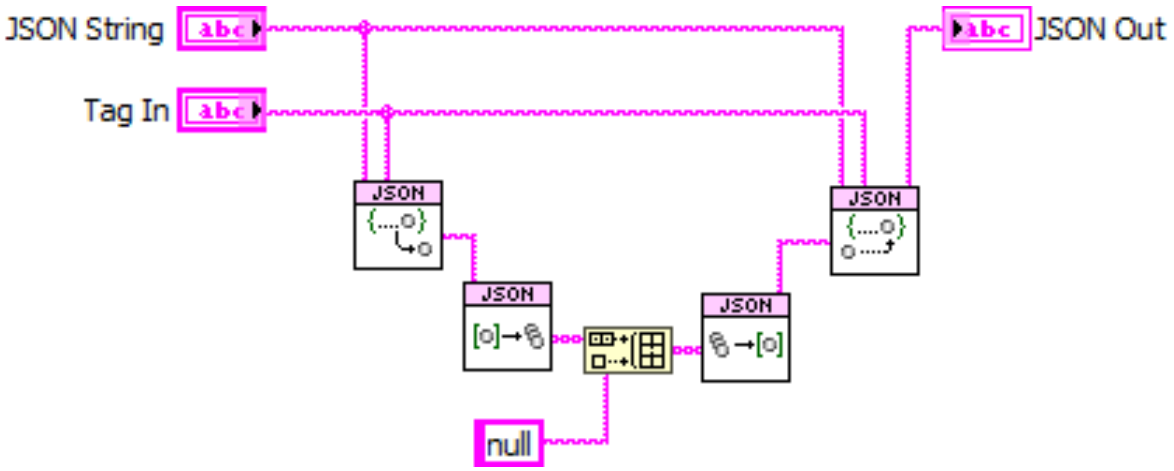
- Comparable to unbundle for clusters
- Comparable to using the JSONtext Find Item

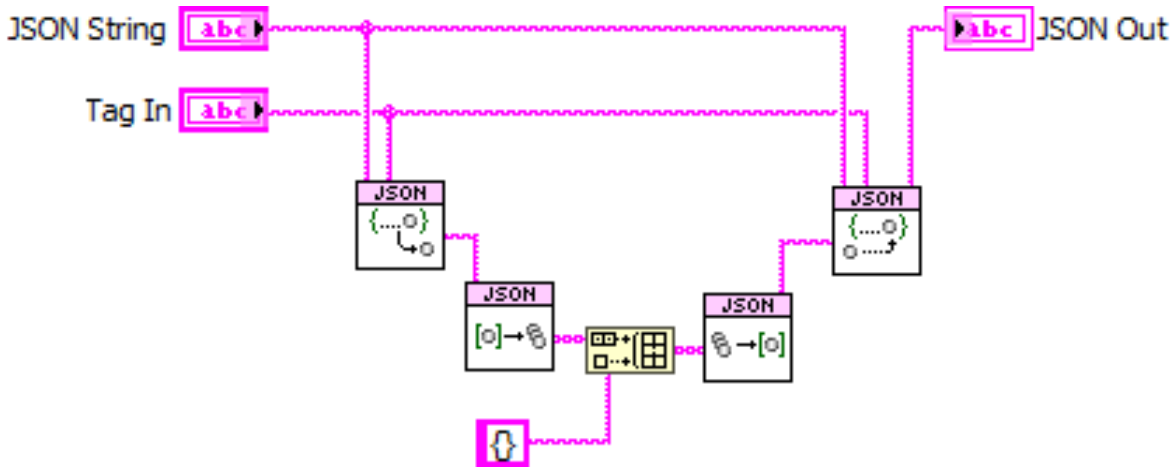
- Useful in queries in combination met WHERE-statement

```
SELECT value FROM JSON_each(json('MY_JSON_STRING'), '$.Actors')
WHERE JSON_extract( value, '$.age' ) > 55;
```

Add empty elements

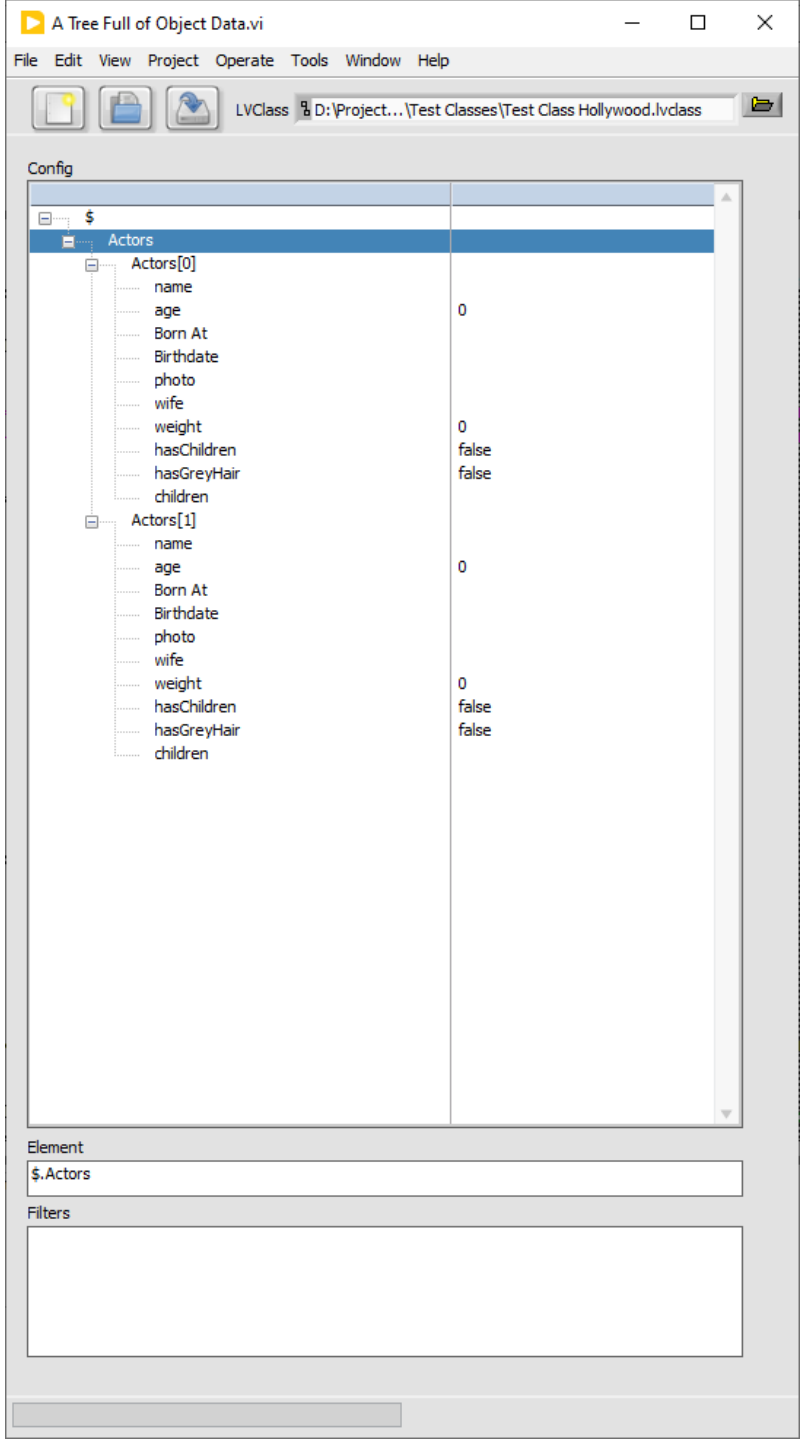
- Use JSONtext to add a default element to an array
- Without knowing the datatype
- Might give error 402844 : JSONtext: Input JSON cannot be converted to requested LabVIEW Data Type





Use Case 1

- JSON Viewer



Use Case 2

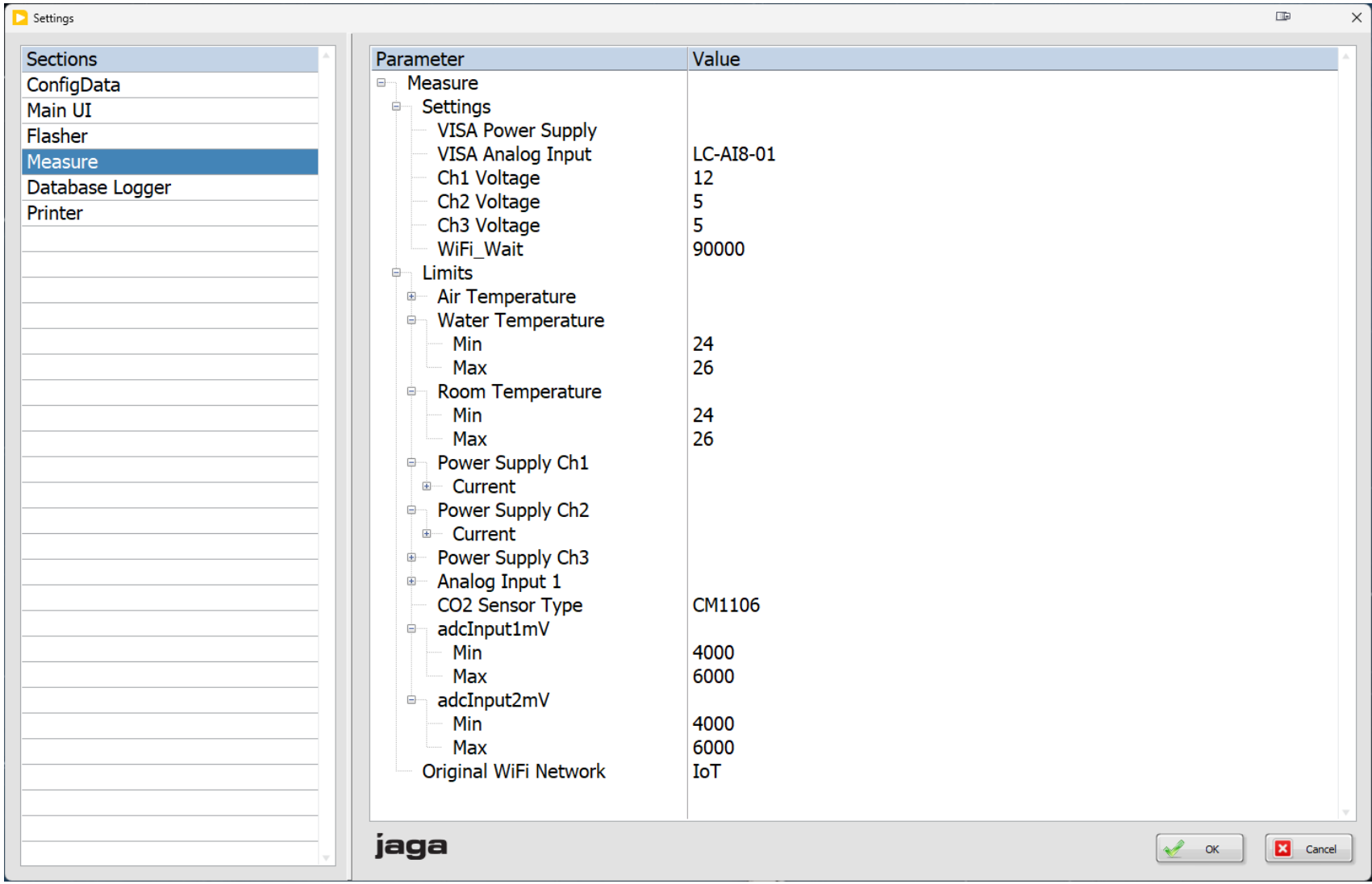
- Result Screen


```
"Air Temperature":{
  "Measured":24.472225189209,
  "Min":24,
  "Max":26,
  "Status":"Pass"
}
```

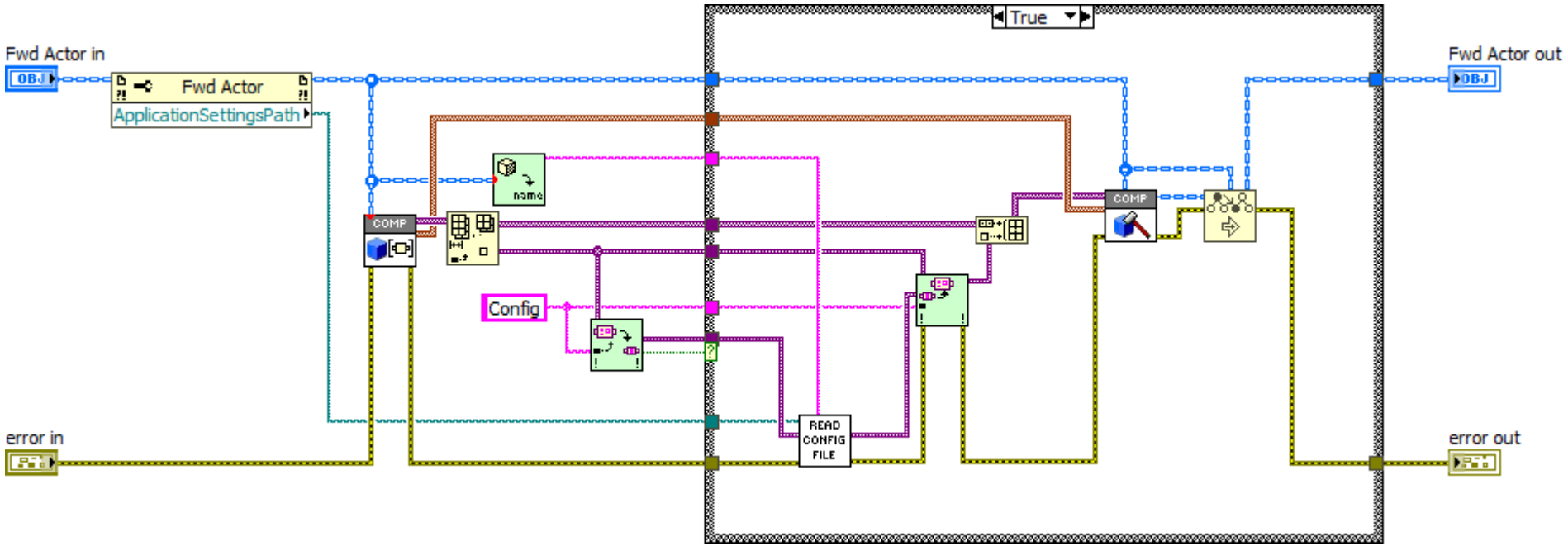

	Minimum	Measured	Maximum
$			
Serial Number		0020240102074211	
MAC Address		08:d1:f9:9f:51:bc	
Pass_Fail		true	
Date		20240701	
Time		090809	
Air Temperature	24	24.472225189209	26
Water Temperature	24	24.5252571105957	26
Room Temperature	24	24.5252571105957	26
CO2 Sensor Type		CM1106	
Current In			
Current In 1			
Current In 2			
Vout	0	1.424	10
adcInput1mV	4000	5032	6000
adcInput2mV	4000	5030	6000

Use Case 3

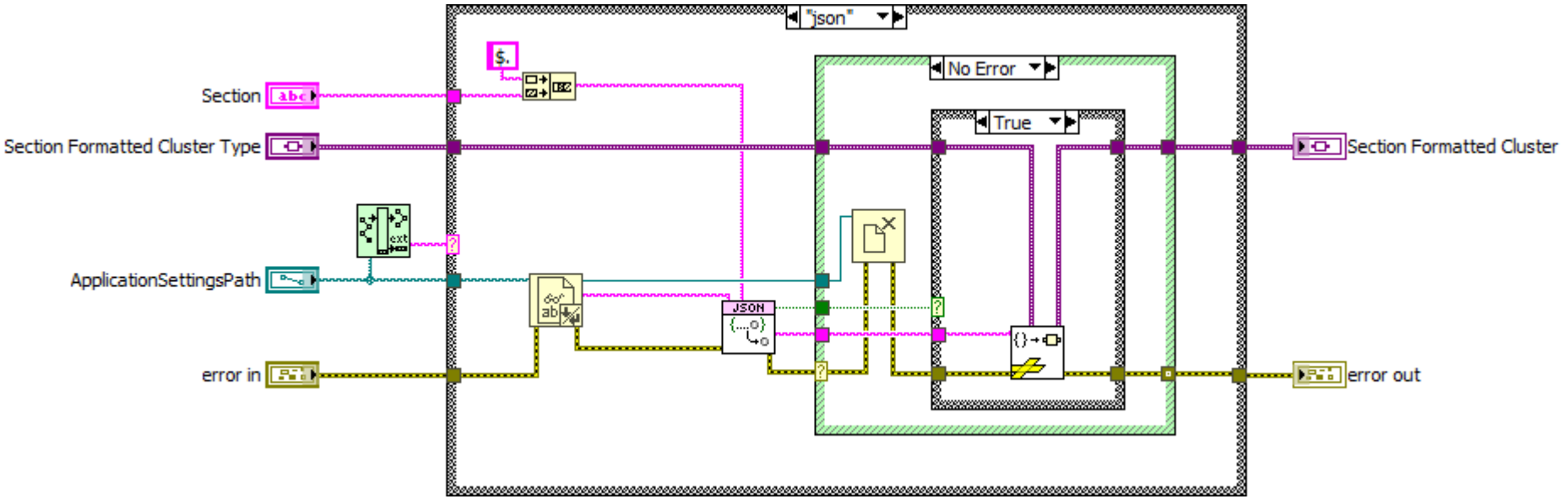
- Config UI



Base Actor


Fwd Actor in
OBJ
Fwd Actor
ApplicationSettingsPath
True
Fwd Actor out
OBJ
name
COMP
Config
COMP
READ
CONFIG
FILE
error in
error out

Read Config from File



Bonus - The other way around

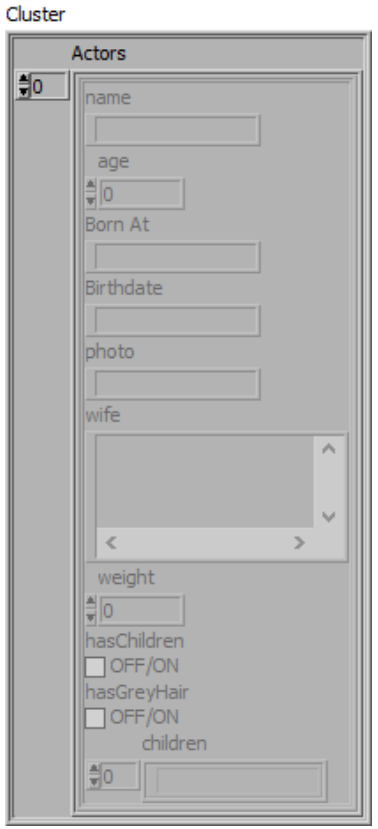
- Creating a LabVIEW cluster typedef from a JSON file


```
{
  "Actors": [{
      "name": "Tom Cruise",
      "age": 56,
      "Born At": "Syracuse, NY",
      "Birthdate": "July 3, 1962",
      "photo": "https://jsonformatter.org/img/tom-cruise.jpg",
      "wife": null,
      "weight": 67.5,
      "hasChildren": true,
      "hasGreyHair": false,
      "children": [
        "Suri",
        "Isabella Jane",
        "Connor"
      ]
    },
    {
      "name": "Robert Downey Jr.",
      "age": 53,
      "Born At": "New York City, NY",
      "Birthdate": "April 4, 1965",
      "photo": "https://jsonformatter.org/img/Robert-Downey-Jr.jpg",
      "wife": "Susan Downey",
      "weight": 77.1,
      "hasChildren": true,
      "hasGreyHair": false,
      "children": [
        "Indio Falconer",
        "Avri Roel",
        "Exton Elias"
      ]
    }
  ]
}
```

Bonus - The other way around

- Creating a LabVIEW cluster typedef from a JSON file

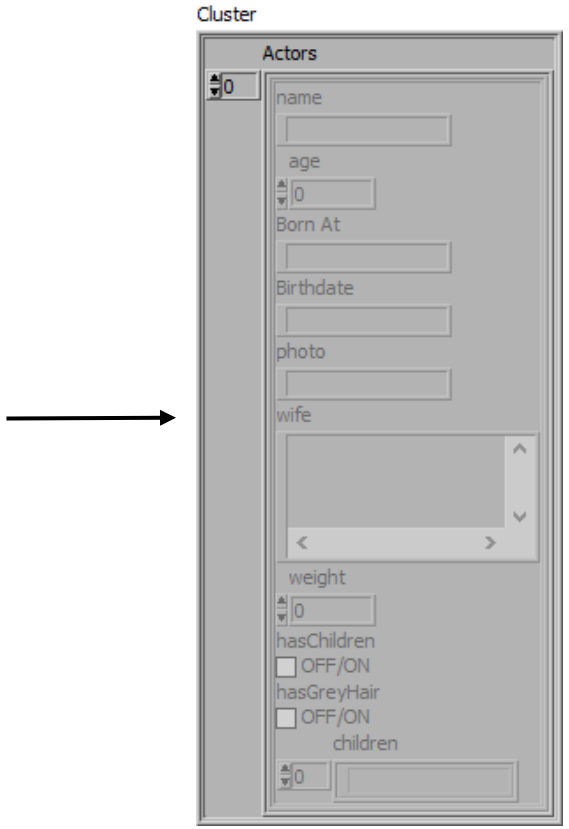
```json
{
  "Actors": [{
      "name": "Tom Cruise",
      "age": 56,
      "Born At": "Syracuse, NY",
      "Birthdate": "July 3, 1962",
      "photo": "https://jsonformatter.org/img/tom-cruise.jpg",
      "wife": null,
      "weight": 67.5,
      "hasChildren": true,
      "hasGreyHair": false,
      "children": [
        "Suri",
        "Isabella Jane",
        "Connor"
      ]
    },
    {
      "name": "Robert Downey Jr.",
      "age": 53,
      "Born At": "New York City, NY",
      "Birthdate": "April 4, 1965",
      "photo": "https://jsonformatter.org/img/Robert-Downey-Jr.jpg",
      "wife": "Susan Downey",
      "weight": 77.1,
      "hasChildren": true,
      "hasGreyHair": false,
      "children": [
        "Indio Falconer",
        "Avri Roel",
        "Exton Elias"
      ]
    }
  ]
}
```

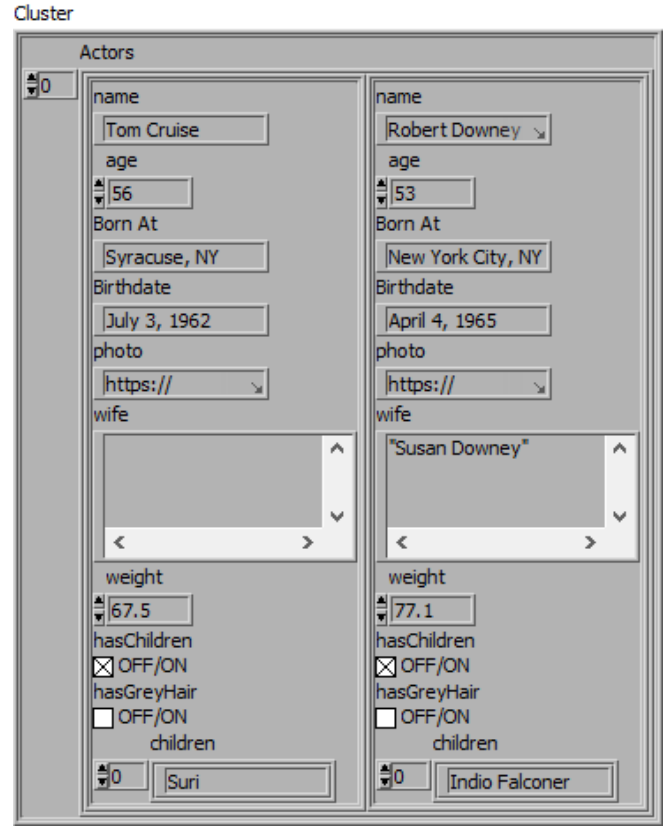
Bonus - The other way around

- Creating a LabVIEW cluster typedef from a JSON file

```
{
  "Actors": [{
      "name": "Tom Cruise",
      "age": 56,
      "Born At": "Syracuse, NY",
      "Birthdate": "July 3, 1962",
      "photo": "https://jsonformatter.org/img/tom-cruise.jpg",
      "wife": null,
      "weight": 67.5,
      "hasChildren": true,
      "hasGreyHair": false,
      "children": [
        "Suri",
        "Isabella Jane",
        "Connor"
      ]
    },
    {
      "name": "Robert Downey Jr.",
      "age": 53,
      "Born At": "New York City, NY",
      "Birthdate": "April 4, 1965",
      "photo": "https://jsonformatter.org/img/Robert-Downey-Jr.jpg",
      "wife": "Susan Downey",
      "weight": 77.1,
      "hasChildren": true,
      "hasGreyHair": false,
      "children": [
        "Indio Falconer",
        "Avri Roel",
        "Exton Elias"
      ]
    }
  ]
}
```

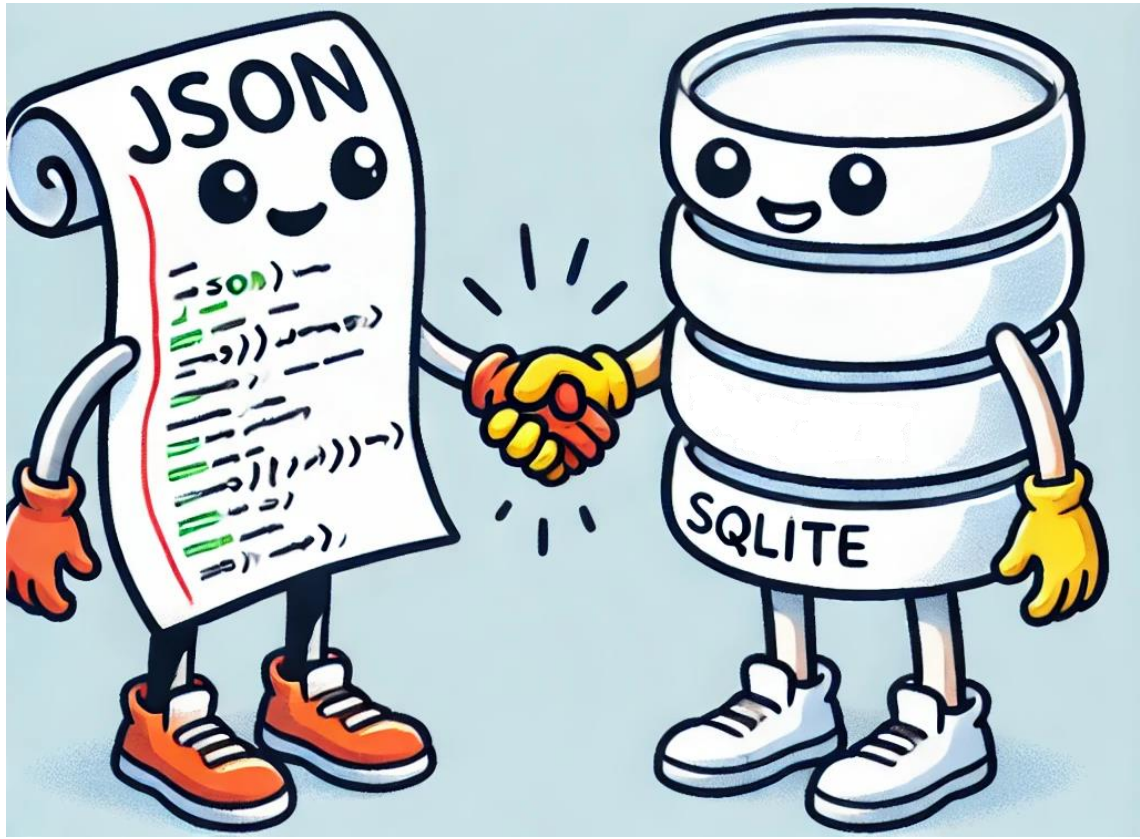


OR



Takeaway

Convert your data into JSON and use SQLite to filter



Packages

- JSONtext by JDP Science
  https://www.vipm.io/package/jdp_science_jsontext/

- SQLite Library by JDP Science
  https://www.vipm.io/package/drjdpowell_lib_sqlite_labview/

- JSONtext Object Serialization by PNR
  https://www.vipm.io/package/pnr_lib_jsontext_object_serialization/

Links

- Application Design Around SQLite - Dr James Powell - GDevCon#2
  https://www.youtube.com/watch?v=i4_l-UuWtPY

- Tree Control – A Little Present
  https://forums.ni.com/t5/Random-Ramblings-on-LabVIEW/Tree-Control-A-Little-Present/ba-p/3486755

- Encrypted SQLite – A Gift
  https://forums.ni.com/t5/Random-Ramblings-on-LabVIEW/Encrypted-SQLite-A-Gift/ba-p/4105019

SQLite Tools

- JSON Validator https://jsonlint.com/
- JSON Path https://jsonpath.com/
- JSON Path Finder https://jsonpathfinder.com/

- DB Browser for SQLite https://sqlitebrowser.org/

- SQLite JSON Functions And Operators https://www.sqlite.org/json1.html

Special thanks to

- Dr James Powell
- Steve Watts
- Pascal Neuperger

jaga CLIMATE DESIGNERS