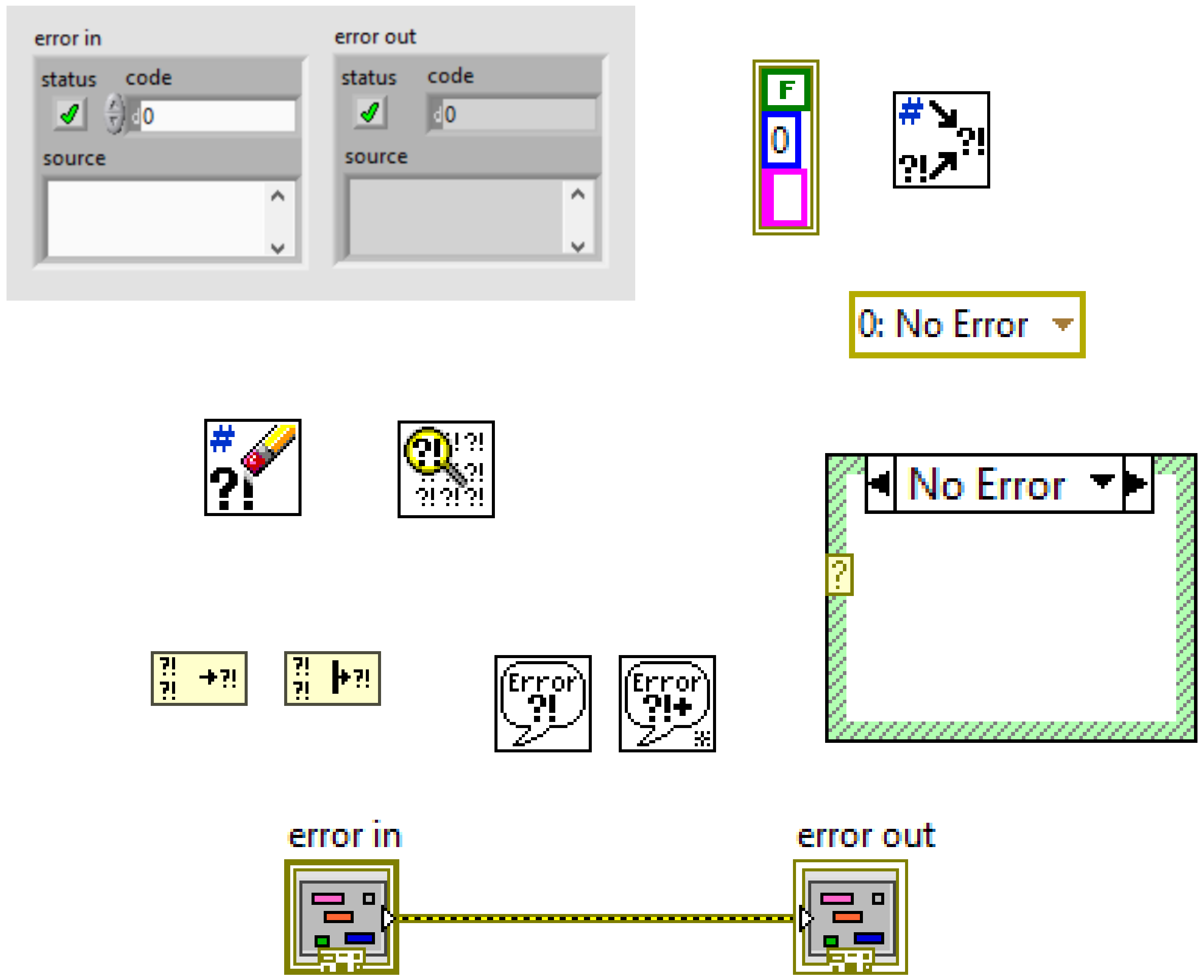
The Errors of Our Ways

Stephen R. Loftus-Mercer

Principal Software Architect


error in
status
code
0
source
error out
status
code
0
source
F
0
0: No Error
No Error
?
Error ?!
Error ?!+
error in
error out

#OurGiantsAreFemale: Dr. Deborah A. Trytten

- My advisor in college
- Professor, University of Oklahoma
- Data structures
- Computer graphics
- Pattern recognition
- Qualitative research into how to teach computer science
  - Automated plagiarism detection
  - How to think about software
  - How to change intuition into understanding that can be shared
  - Elimination of structural barriers to women and minorities
  - How to tame “smart cowboys”

In the real world, you will mostly work in teams. Get used to it.

Dr. Deborah A. Trytten

University of Oklahoma, Computer Science

There is nothing special about error data.

Error data is the most special data in any application.

What is an error?

(in software specifically)

Game: Error or No Error?

- Each of the following functions performs a task.
- For each call, it may either return a result or an error.
- Are the functions behaving correctly?

Function: Wash Dishes

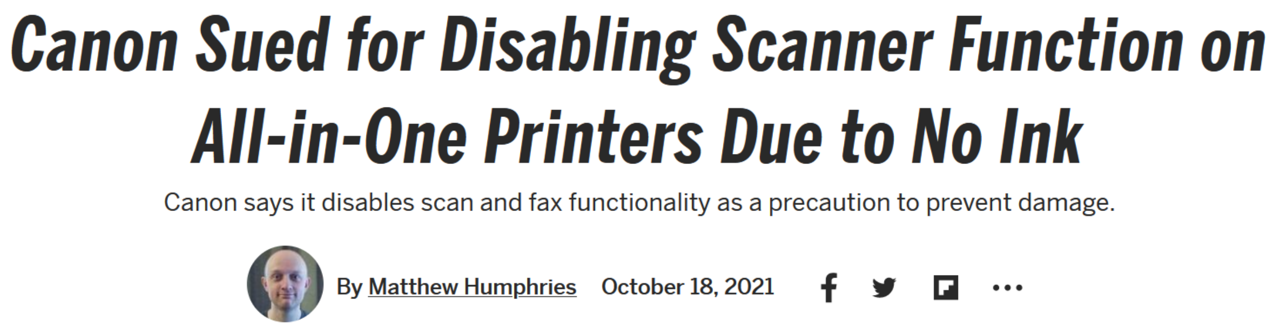
- Dishwasher takes in a load of dishes. Wash cycle takes about 2 hours.

Situation	Error or No Error?
All dishes cleaned in 118 minutes	✓ No Error
Water outage	✗ Error: No water
Electrical failure	✗ Error: No power
All dishes cleaned in 63 minutes	✓ No Error
Started run without detergent	✗ Error: No detergent

Function: Scan Document

- Canon All-In-One Printer-Scanner-Fax

Situation	Error or No Error?
Out of printer ink	☒ Error: Cannot scan document



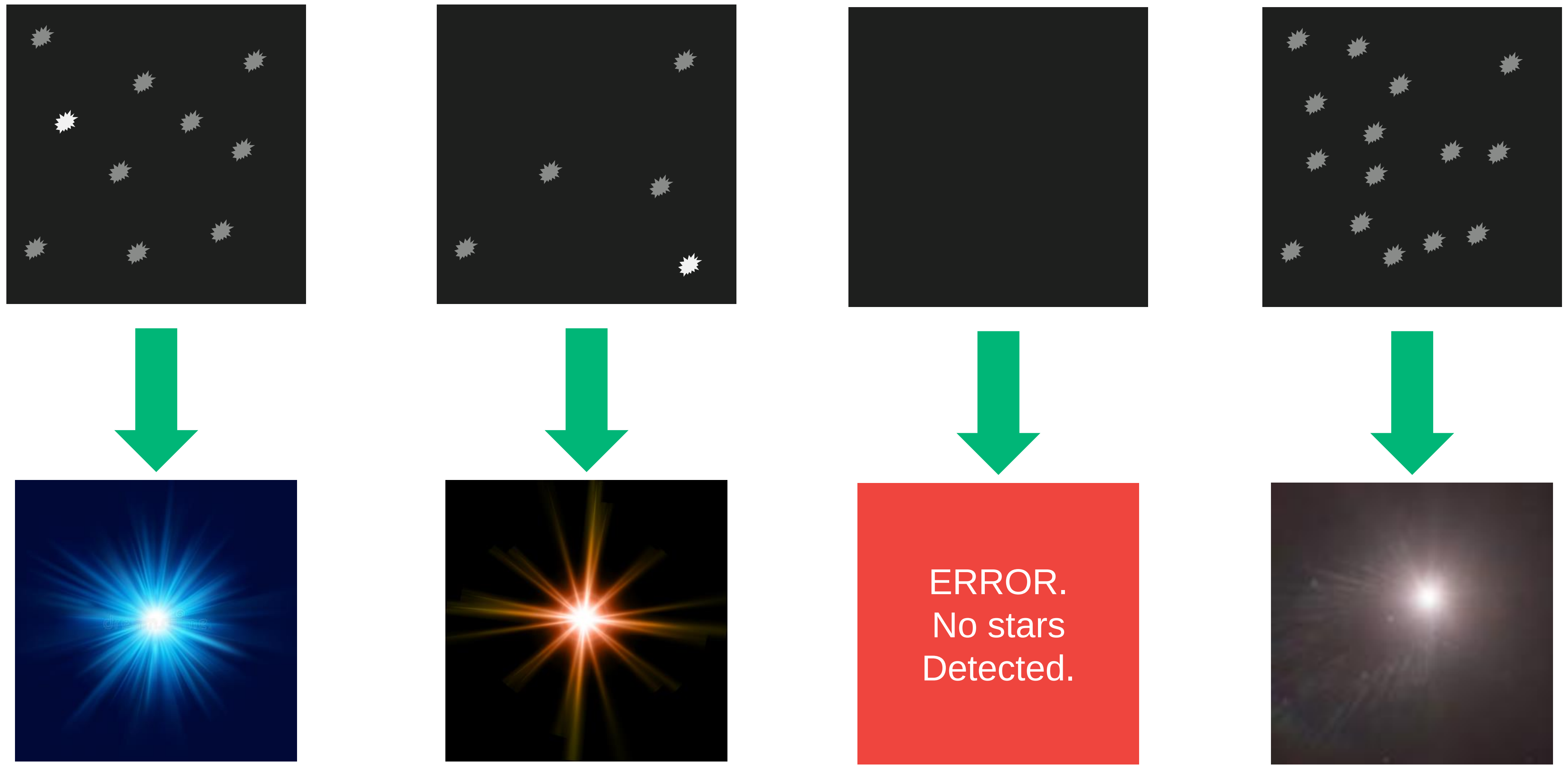
Canon Sued for Disabling Scanner Function on All-in-One Printers Due to No Ink

Canon says it disables scan and fax functionality as a precaution to prevent damage.

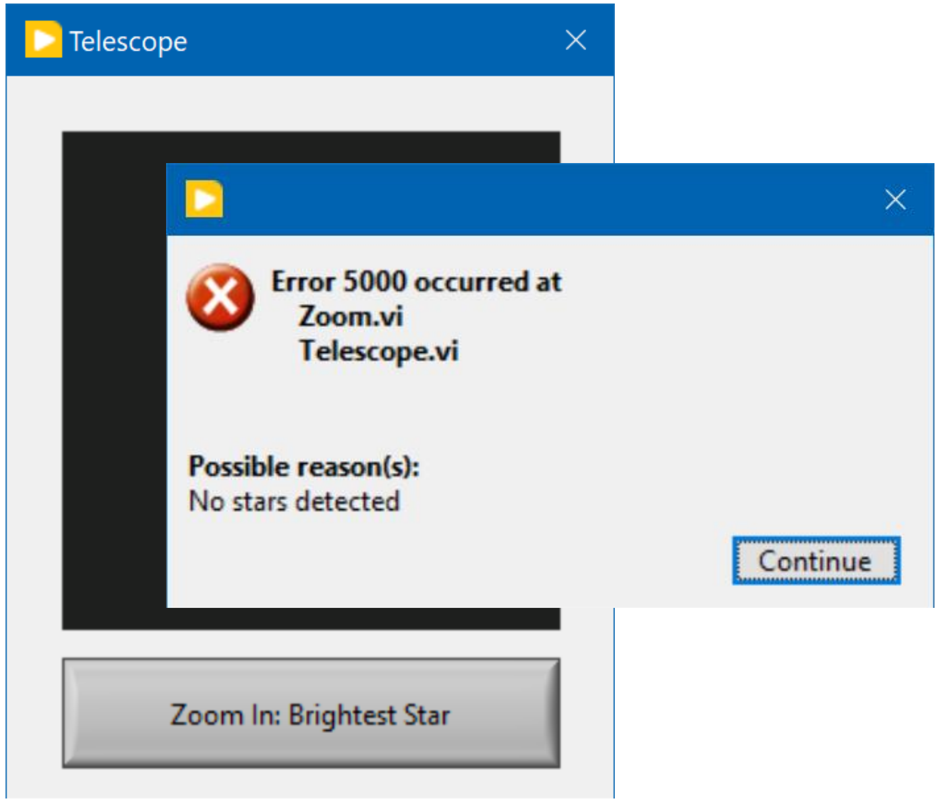
By Matthew Humphries October 18, 2021

https://www.pcmag.com/news/canon-sued-for-disabling-scanner-function-on-all-in-one-printers-due-to

Function: Zoom in on brightest star

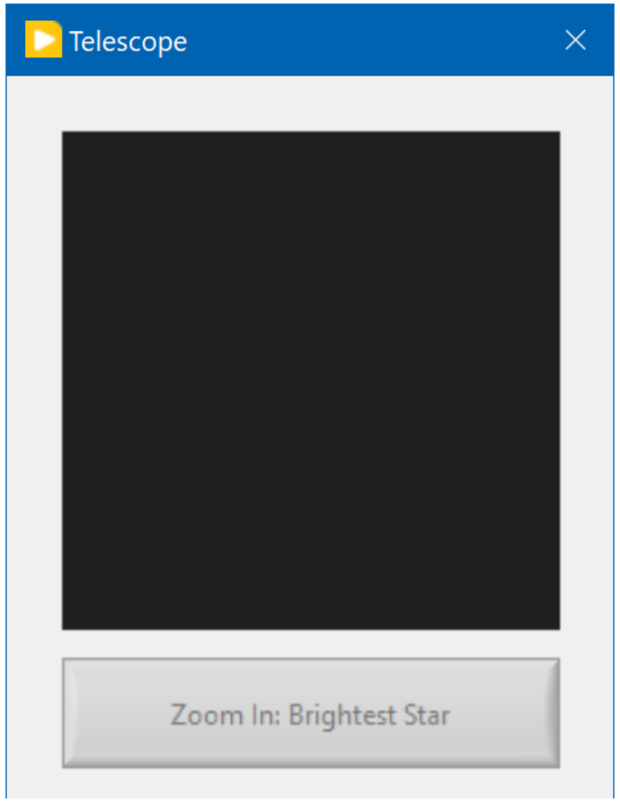


Caller

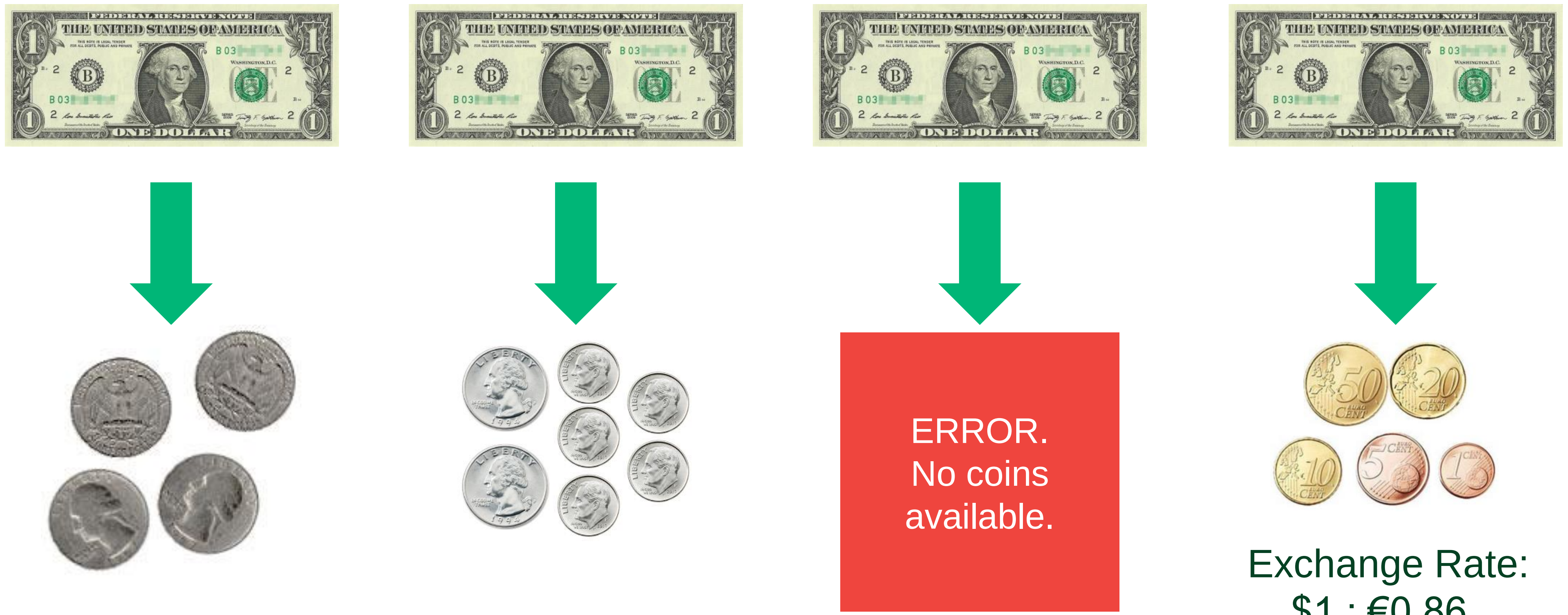


Error 5000 occurred at
Zoom.vi
Telescope.vi

Possible reason(s):
No stars detected



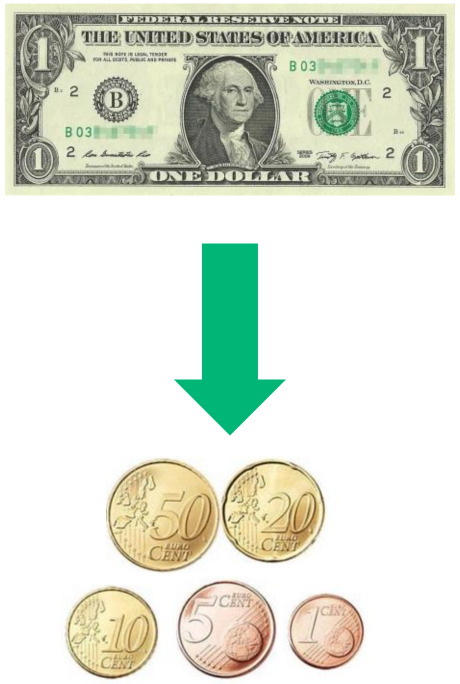
Function: Break bill into coins



Exchange Rate:
$1 : €0.86

Function: Break bill into coins

(long delay)



Exchange Rate:
$1 : €0.86

Caller

With Possible Errors

```
Loop
    Call “Break Into Coins”
    If “no error”, exit loop.
    After 3 tries, return error.
Take first coin.
Give to referee.
```

Without Possible Errors

```
Call “Break Into Coins”
Take first coin.
Give to referee.
```

Game: Do these functions behave correctly?

Yes. By definition.

What constitutes an error...

... depends *entirely* upon context.

Errors are only errors…

… from the sender or subVI point of view.

Working Definition of “Error”

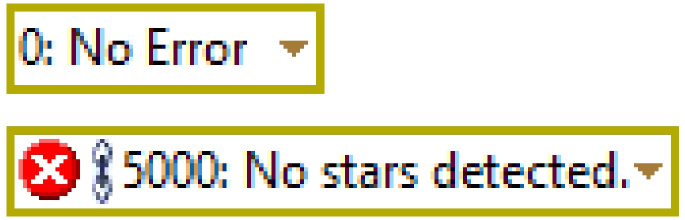
An error is an indication that a function could not complete its assigned task.

LabVIEW Error Clusters

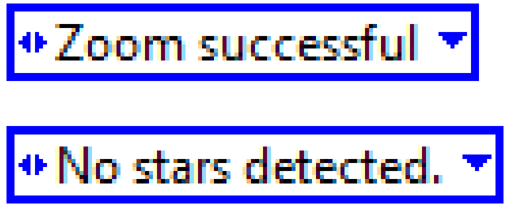
Why an error cluster?

If errors are so context dependent, why not have every node return whatever data best describes its status?

ERROR.
No stars
Detected.

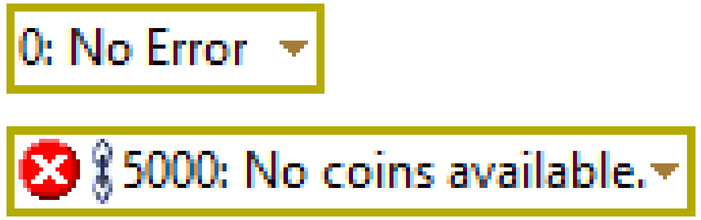

0: No Error

5000: No stars detected.

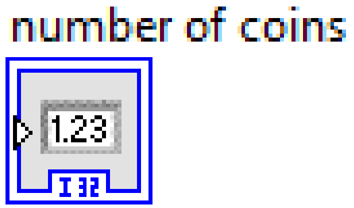

Zoom successful

No stars detected.

ERROR.
No coins
available.


0: No Error

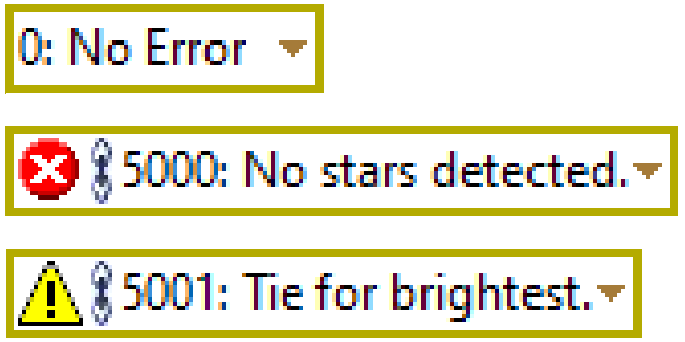
5000: No coins available.



Why an error cluster?

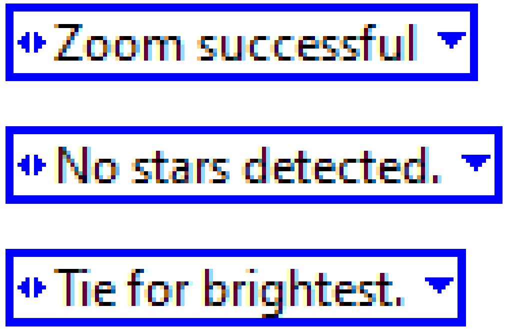
If errors are so context dependent, why not have every node return whatever data best describes its status?

ERROR.
No stars
Detected.


0: No Error

5000: No stars detected.

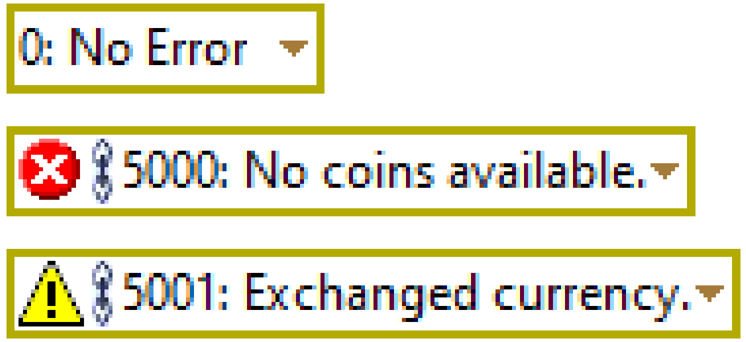
5001: Tie for brightest.


Zoom successful

No stars detected.

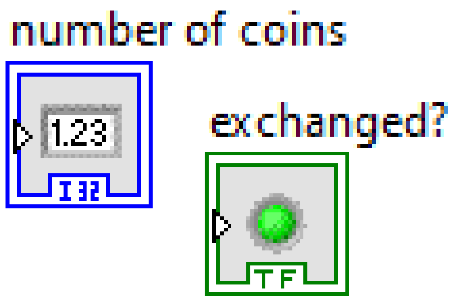
Tie for brightest.

ERROR.
No coins
available.


0: No Error

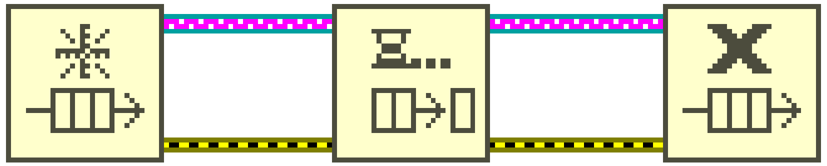
5000: No coins available.

5001: Exchanged currency.



Why an error cluster?

Unified type benefits callers.



We can pack a lot of metadata into that cluster.

The computer knows what went wrong.

Be sure to tell the user.

Roy Faltesek

My first manager at NI

Explain The Errors

- **Unhelpful:**
  Error code 7: File not found.
- **Useful:**
  Error code 7: File not found.
  Path: c:\temp\configuration.json

Errors Change Explanations As They Ascend Call Chain

- Sometimes you want the low-level detail:
  - **Function:** Execute Battleplan.vi
    Unhelpful error message: “Attack failed.”
    Useful error message: “Could not find horseshoe nail.”

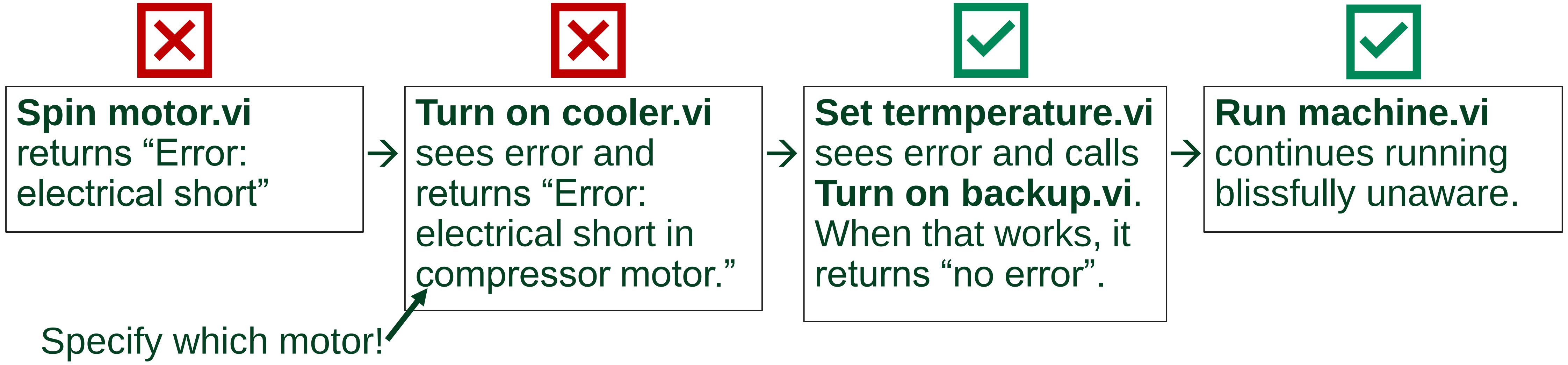
For Want of a Nail
For want of a nail the shoe was lost.
For want of a shoe the horse was lost.
For want of a horse the rider was lost.
For want of a rider the message was lost.
For want of a message the battle was lost.
For want of a battle the kingdom was lost.
And all for the want of a horseshoe nail.

- Sometimes, you do not:
  - **Function:** Run Nuclear Power Plant.vi
    Unhelpful error message: “Temperature sensor overload. Replace sensor.”
    Useful error message: “Reactor meltdown in progress. Evacuate.”

Errors Change Explanations As They Ascend Call Chain

Call chain

- Run machine.vi
- Set temperature.vi
- Turn on cooler.vi
- Spin motor.vi



✗ **Spin motor.vi** returns “Error: electrical short”

→

✗ **Turn on cooler.vi** sees error and returns “Error: electrical short in compressor motor.”

→

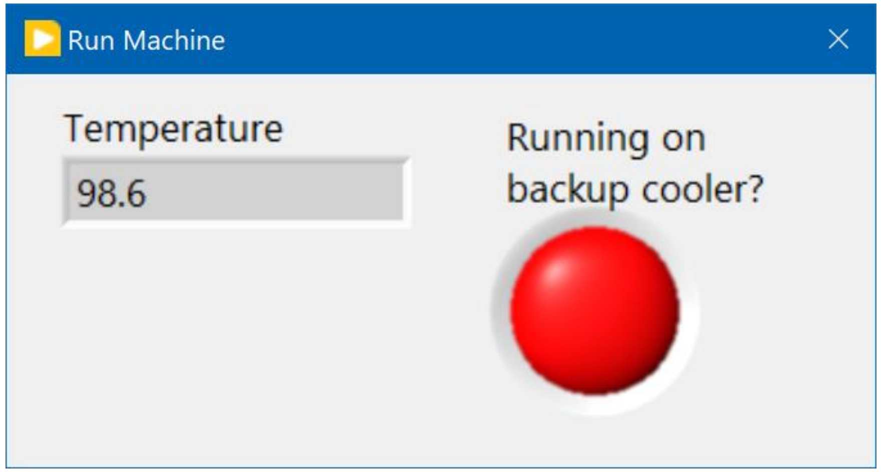
✓ **Set termperature.vi** sees error and calls **Turn on backup.vi**. When that works, it returns “no error”.

→

✓ **Run machine.vi** continues running blissfully unaware.

Specify which motor!

Small Changes in Requirements



✗

Spin motor.vi returns “Error: electrical short”

→

✗

Turn on cooler.vi sees error and returns “Error: electrical short in compressor motor.”

→

✓ Backup Activated

Set termperature.vi sees error and calls **Turn on backup.vi**. When that works, it returns “Backup Activated” status.

→

✓

Run machine.vi continues running blissfully unaware.

Is “Backup Activated” a warning?

✓ + Backup Activated $\overset{?}{=}$ ⚠

Maybe? But I wouldn’t.

Backup Activated

Fan Rattling

Filter Needs Changing

1. Maybe need multiple messages simultaneously.
2. Ever try to parse an error cluster string?

Errors Are Not Special

- An error cluster is just a cluster.
- You can pack information into it (for good or for ill).
- You can have subVIs return status in addition or instead of an error cluster.
- It’s just a status message.

Error Driven Design

Errors Are The Most Special Data Type

- Error information is likely to create data dependency from the bottom of your app to the top.
- Error code paths are often under specified.
- Error code paths are often harder to test than non-error code paths.

“Let’s start a runaway nuclear chain reaction and see if error handling catches it!”

Errors Are The Most Special Data Type

- Errors require a specific data type that is consistent across functions, across domains.
    - Errors become a language feature, not a library feature.
- Error handling develops custom syntax patterns because it is used *continuously*.
- A change in error reporting requirements can change entire application architectures.

“Why does the radio module know about the tire balance sensors in this vehicle?!”
“That’s the display panel for user status. It needed an LED to show out of balance.”

Aspects of Error Handling

Five distinct aspects that all get bundled under “handling”:

1. Error Generation
2. Error Propagation
3. Error Response
4. Error Display
5. Error Logging

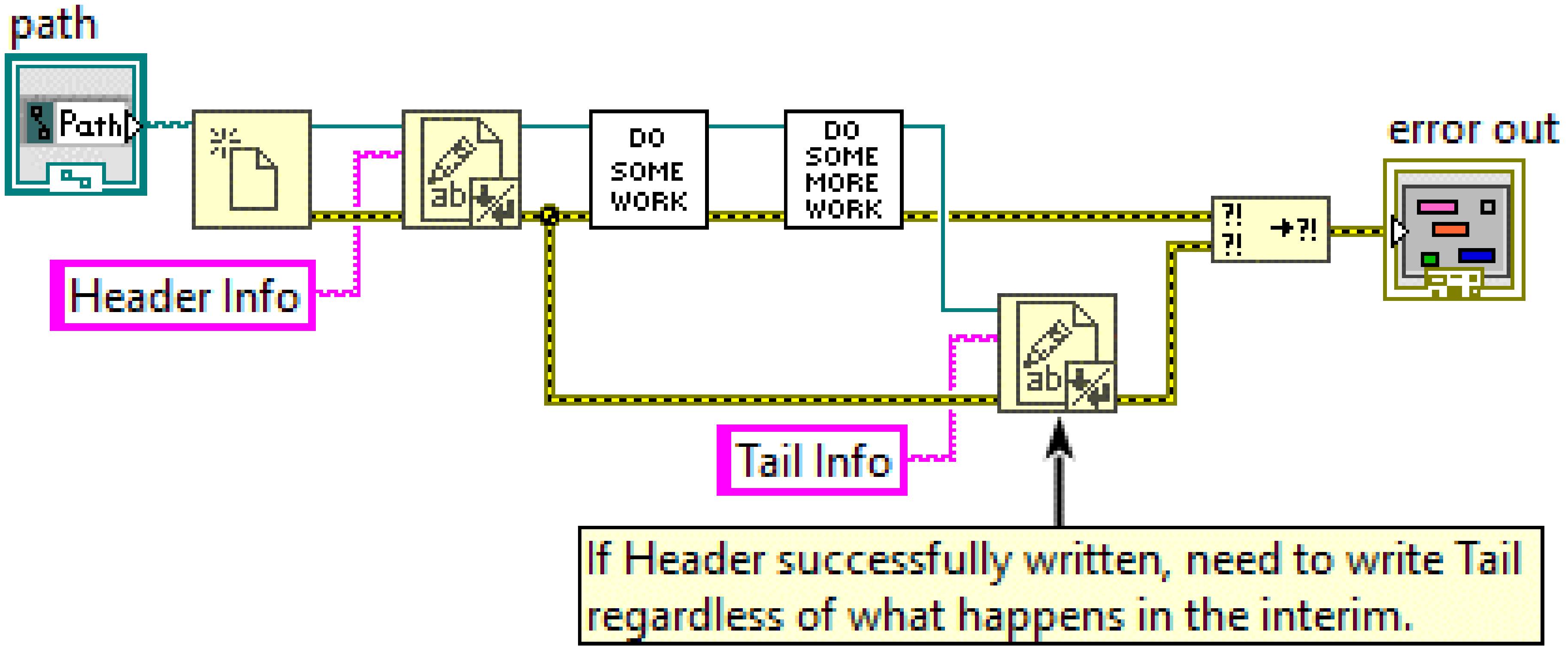
A "Need to Know" Basis

- Design your application around your error propagation.
- If Module X fails, which other modules care?
- By focusing on the error cases, we can avoid creating unsolvable edge and corner cases.

failing

"Let's just get something ~~working.~~"

Apply Error Design Even At Diagram Level


path
Path
Header Info
DO SOME WORK
DO SOME MORE WORK
Tail Info
error out
If Header successfully written, need to write Tail regardless of what happens in the interim.

Consider Cases We Have Seen Earlier

- The telescope UI did not care about the errors from zoom because it was pre-checking inputs.
- The referee function infinitely waited unless we propagated errors from break coins.
- The kingdom could not be saved without deep knowledge about horseshoes.
- Canon would not be able to rake in $ from ink sales if it didn’t propagate errors to scanner.

Error Recommendations

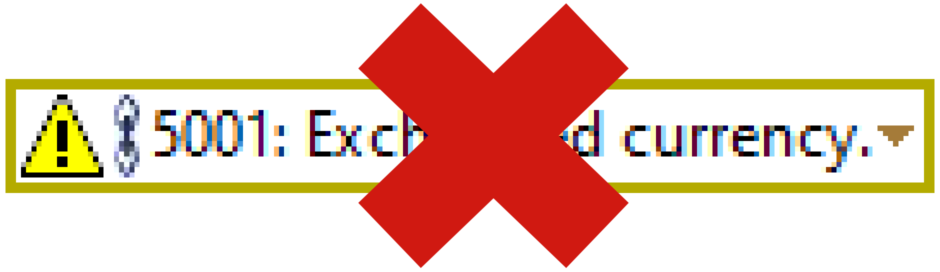
Recommend: Eliminate Excess Error Terminals

- Don’t bother with “error in” if the function does not have effects beyond computation.
- Don’t bother with “error out” if the function encodes error in its data (e.g., NaN for double)
- **You can have “error out” without “error in”!**
  - (especially on protected scope dynamic dispatch VIs)

Recommend: Do not use warnings

- Warnings are hard to be aware of (connector pane doesn't alert you "here be warnings").
- Warnings are easily dropped in the data flow.
- Propagating warnings can create extra data flow dependencies.
- **Warnings are generally local care abouts only.** Upgrade to error or ignore. Maybe log.

Use status outputs instead.



Recommend: Recontextualize at Module Boundaries

- Module boundaries
  - Returning error from VI inside a library to caller outside the library.
  - Messaging between parallel operations.
  - Sending error across the network.

"Should this function translate the low-level error into something its caller will know about?"

Questions