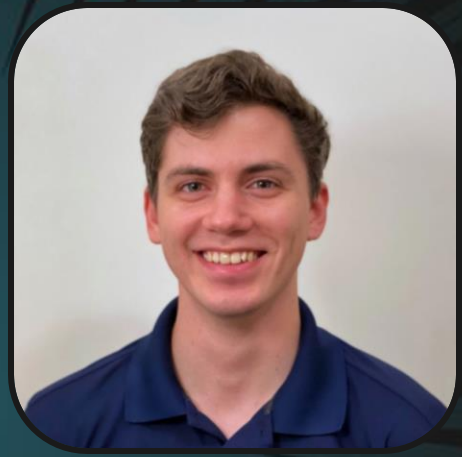

IGNITION COMMUNITY CONFERENCE 2024®
BREAKTHROUGH
TO YOUR NEXT BIG IDEA!

Optimizing Load Time in Ignition Perspective

DMC

ICC 2024



Elizabeth Reed

Senior Manager, SCADA and MES

DMC, Inc.

Casimir Smith

Project Engineer

DMC, Inc.

DMC

ICC 2024

Types of Performance

Perceived Performance

How long do pages feel to load

Responsiveness

How long between user action and result

Resource Usage

How much RAM or CPU is being used

DMC

ICC 2024

So how do we prevent this from happening?

DMC

ICC 2024

Improving Perceived Performance

Fundamentals

Reduce amount of data and calculation

Reduce layout recalculations

Reduce initial load actions

Reduce heavyweight components

Advanced techniques

Analyze loading

Hide loading

DMC

ICC 2024

Fundamentals

Reduce amount of data and calculation

Reduce layout recalculations

Reduce initial load actions

Reduce heavyweight components

Advanced techniques

Analyze loading

Hide loading

DMC

ICC 2024

Binding Efficiency

Binding Efficiency Ranking

★★★ Direct tag binding

★★★ Indirect tag binding

☆☆☆ tag() expression

☆☆☆ runScript() expression

Transform Efficiency Ranking

★★★ Format transform

★★★ Map transform

★★☆ Expression transform

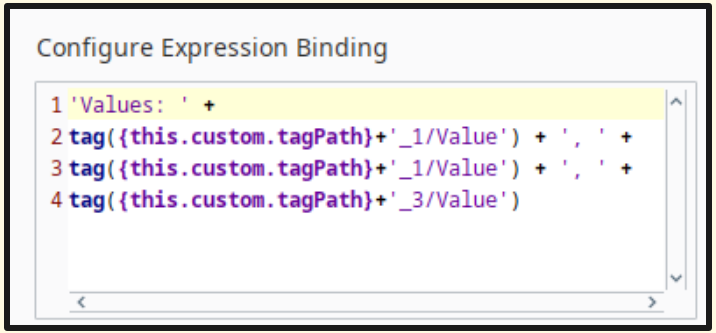
★☆☆ Script transform

DMC

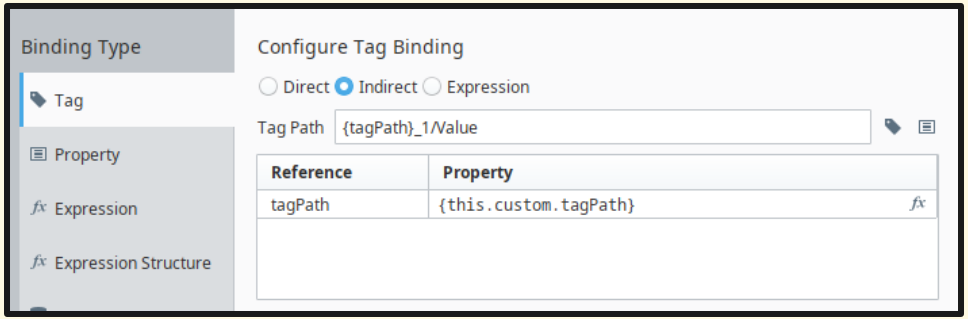
ICC 2024

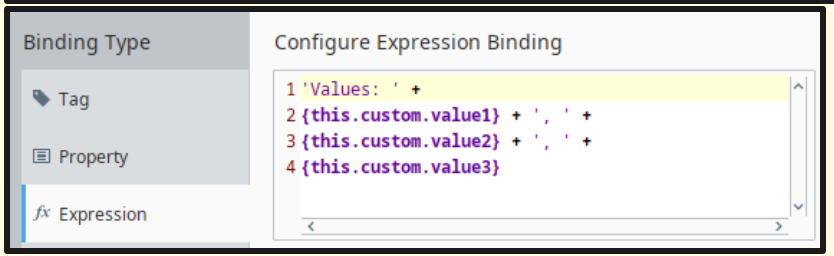
Binding Efficiency

Avoid use of tag()



Instead, use multiple indirect tag bindings





DMC

ICC 2024

Query Efficiency

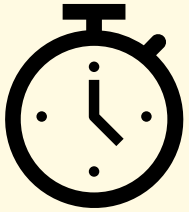
- Always include a range limiter on queries that select from large tables
  - Time-based or count-based
- Avoid polling
  - Use a refresh button that calls refreshBinding() instead
- Use named queries with caching enabled
- Use database tools to troubleshoot query performance

DMC

ICC 2024

Scripting Efficiency

- Don’t script unless you need to
- Avoid duplicating logic in a loop
- Use bulk tag reads instead of one at a time



Bulk tag reads are up to 10x faster

```python
def readTags(folder,tagNames):
    """
    Package tag paths, then perform a single
    readBlocking() call, then grab the values
    """
    paths = ['{}/{}'.format(folder,tag) for tag in tagNames]

    qvals = system.tag.readBlocking(paths)

    tagValues = [qval.value for qval in qvals]
    return tagValues
```

DMC

ICC 2024

Fundamentals

Reduce amount of data and calculation

Reduce layout recalculations

Reduce initial load actions

Reduce heavyweight components

Advanced techniques

Analyze loading

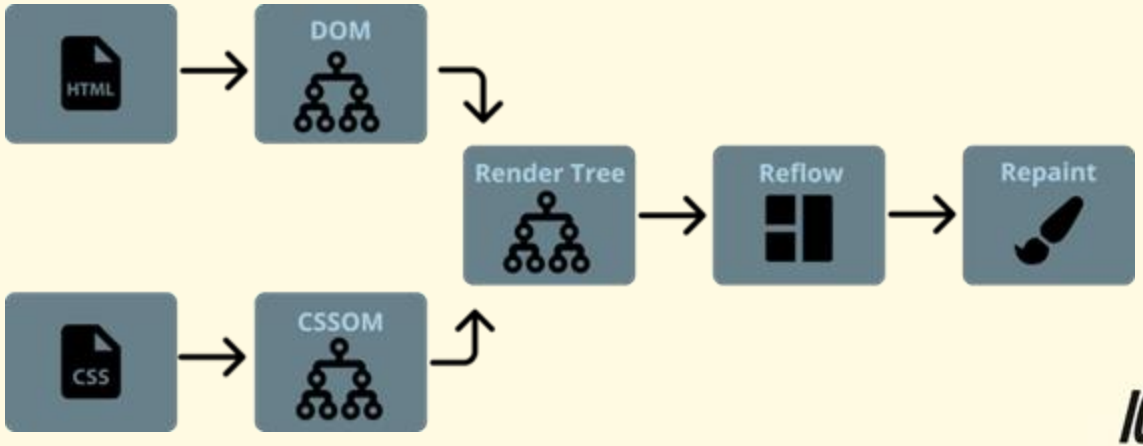
Hide loading

DMC

ICC 2024

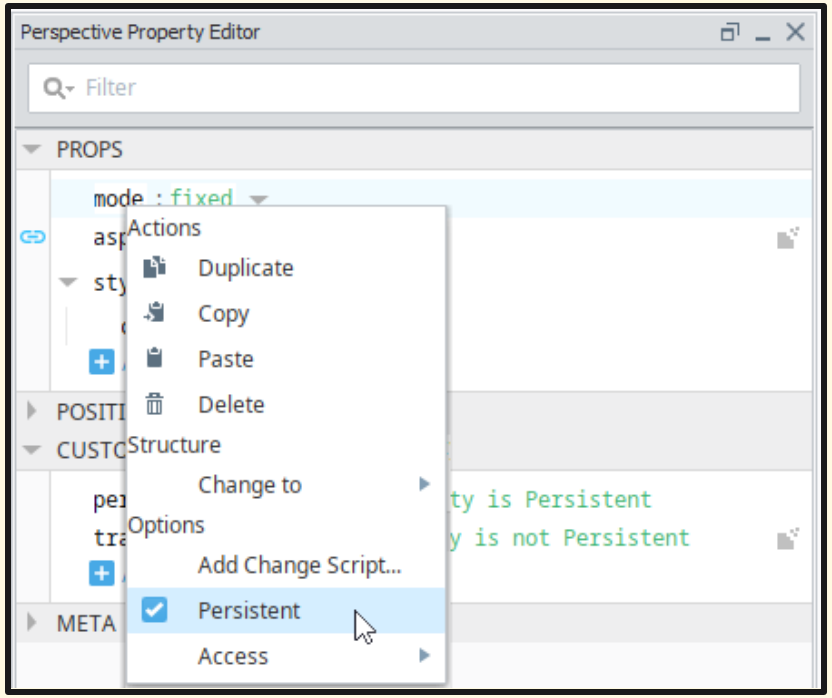
Reduce Reflow and Repaint

- **Reflow:** recalculating position and geometry
  - Expensive process
- **Repaint:** changes that affect visibility, but not layout
- Avoid bindings that alter page layout



Preset Layout with Persistence

- Change a bound property to “persistent” to save the starting value instead of initializing to null.
- Use persistence to set starting layout.
- Use persistence to avoid red overlays on initial load.

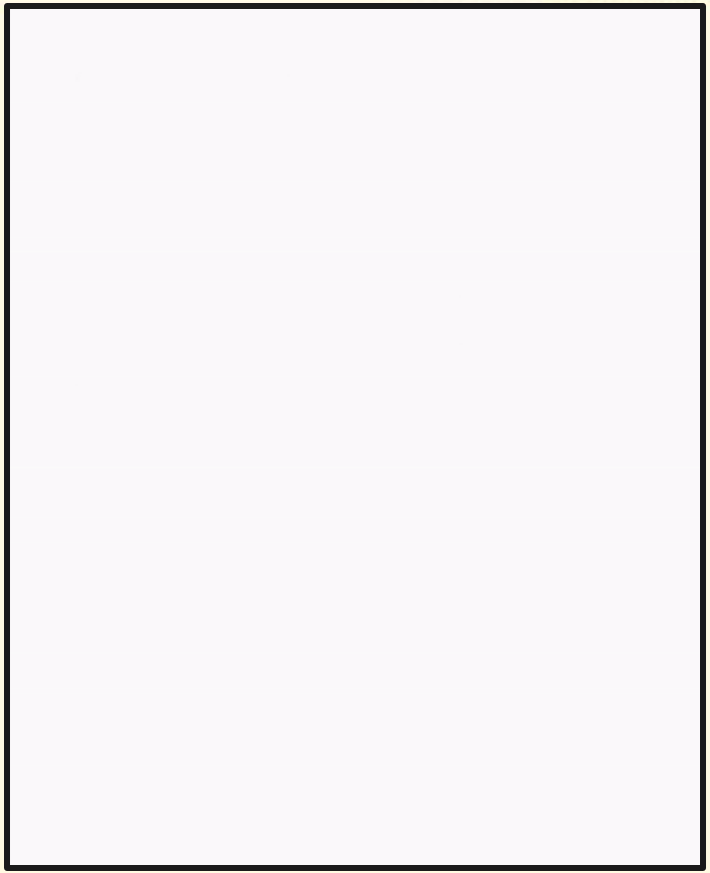


DMC

ICC 2024

Nested Embedded Views

- Avoid nesting more than 3 layers deep
- Pass in tag paths as parameters, then use indirect tag bindings
- Try both with-parent and after-parent loading



DMC

Fundamentals

Reduce amount of data and calculation

Reduce layout recalculations

Reduce initial load actions

Reduce heavyweight components

Advanced techniques

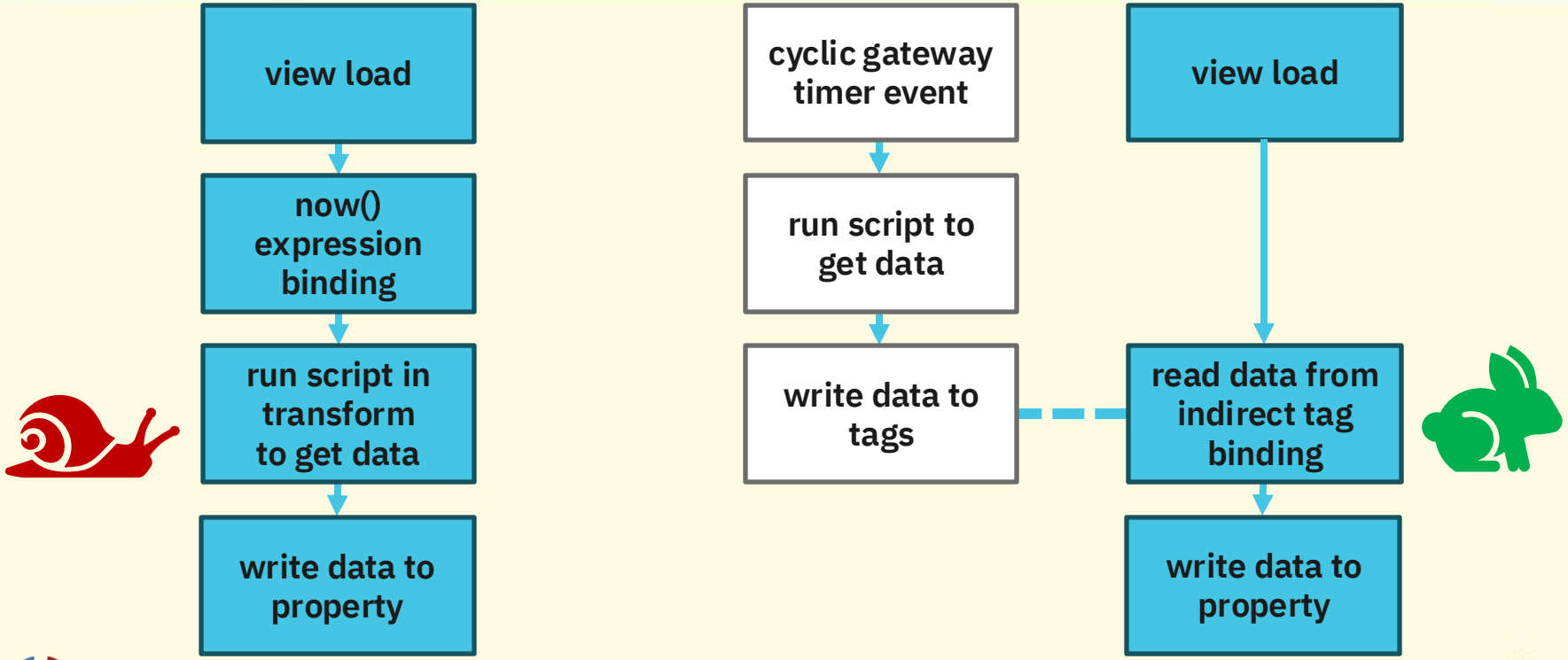
Analyze loading

Hide loading

DMC

ICC 2024

Move Calculations to Gateway Scope



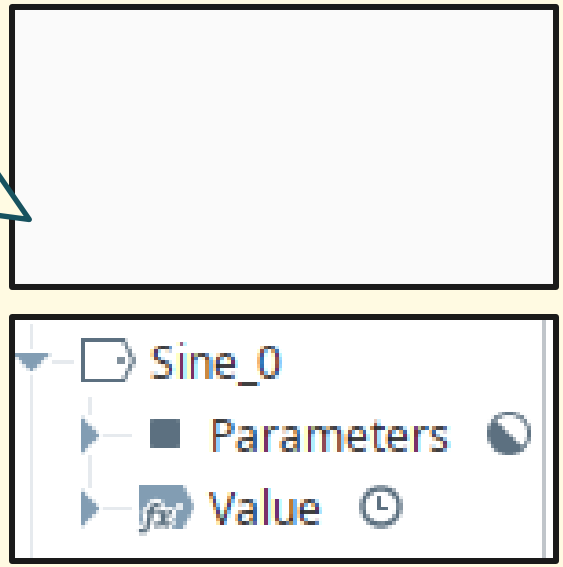
DMC

ICC 2024

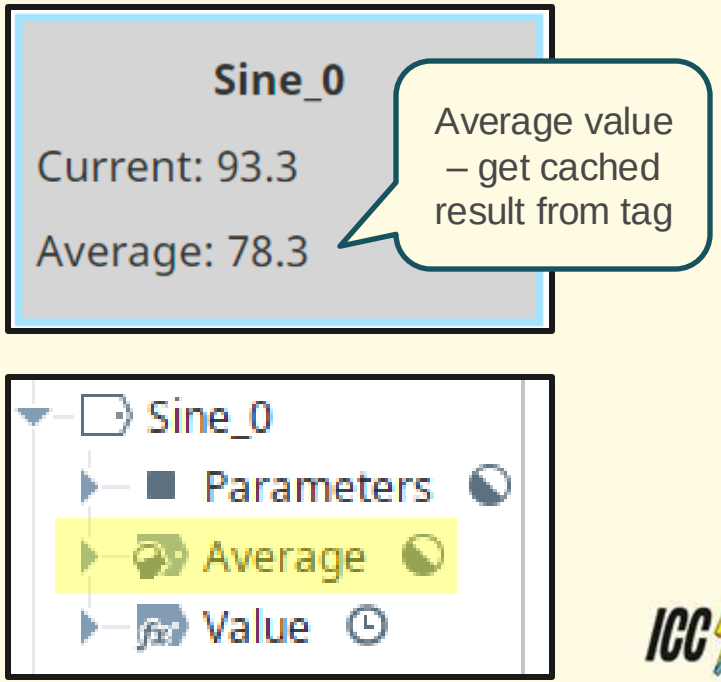
Move Calculations to Gateway Scope

Run Script on Screen: Slow Load

Average value – run script on a client screen



Precalculate Script Result: Instant Load



Note: for both of these instances the "current" is a tag binding

DMC

ICC 2024

Fundamentals

Reduce amount of data and calculation

Reduce layout recalculations

Reduce initial load actions

Reduce heavyweight components

Advanced techniques

Analyze loading

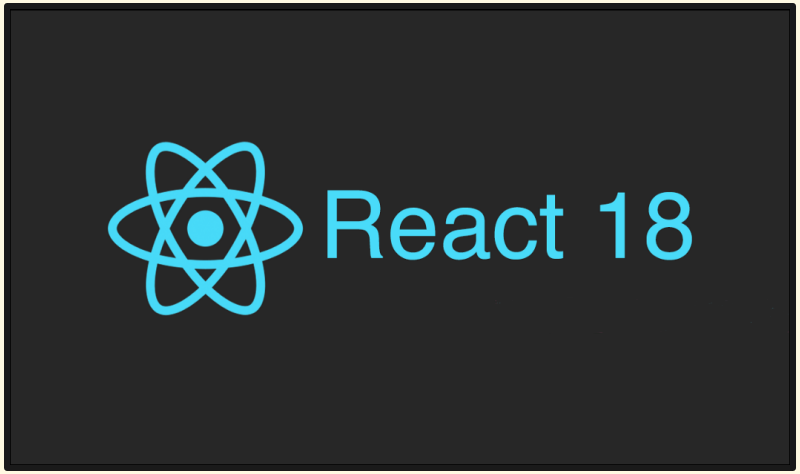
Hide loading

DMC

ICC 2024

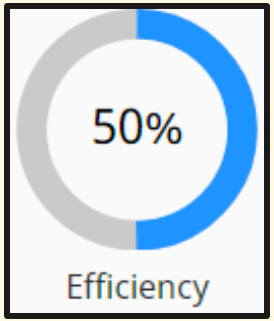
Ignition Version

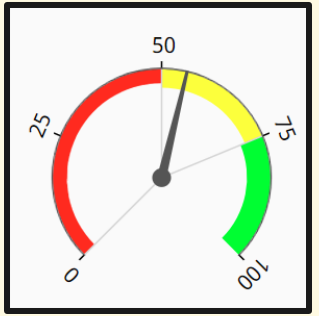
- If you are before 8.1.31, upgrade Ignition

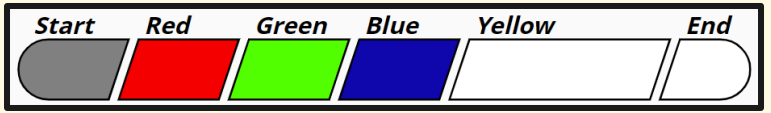


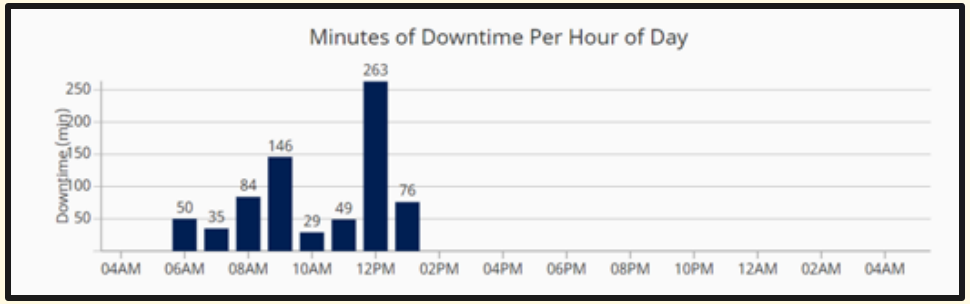
Custom Lightweight Components

- Embed SVG graphics for cases with many repeated components
- Ignition Exchange has example gauges and charts
- Heavyweight components
  - **XY Chart**
  - Gauge & Pie Chart
  - Markdown







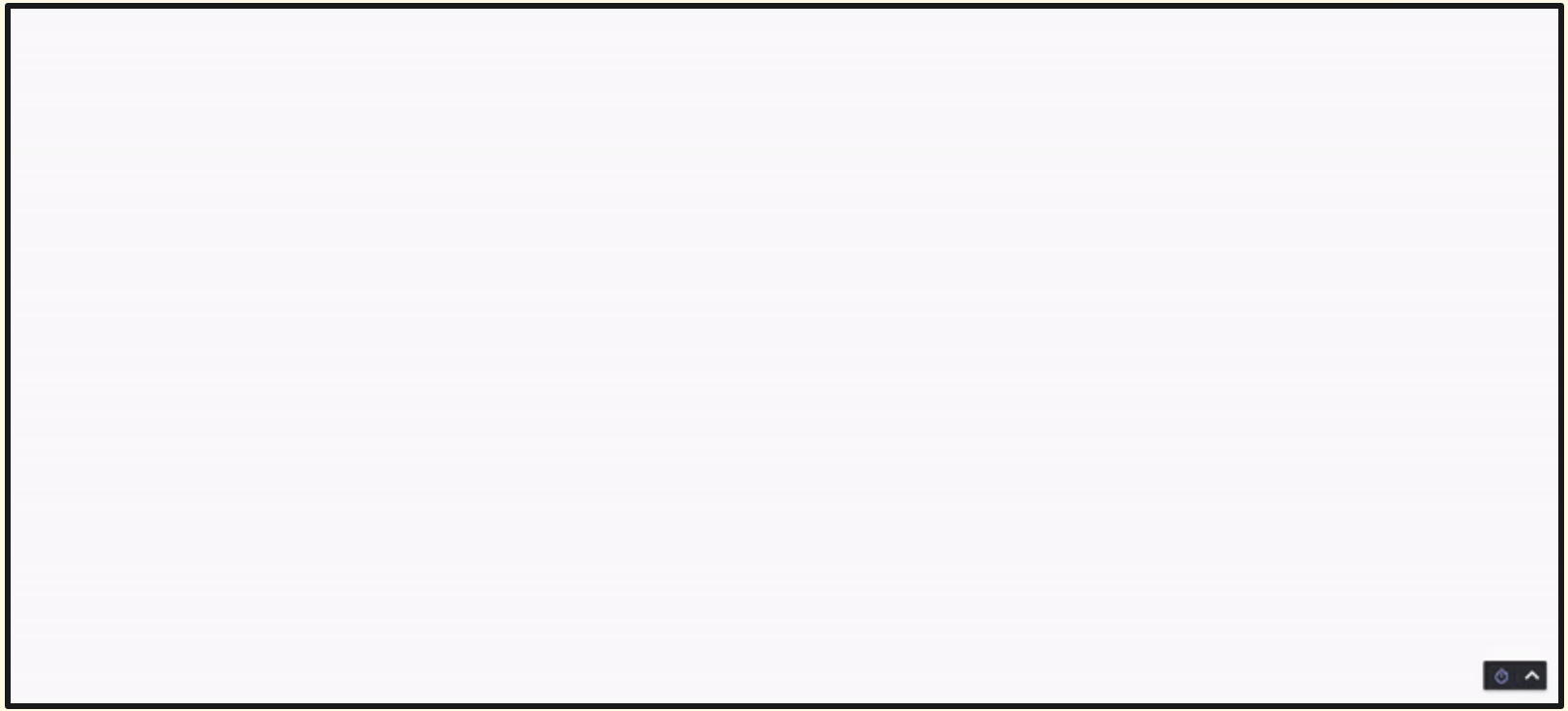


Ignition Exchange

DMC

ICC 2024

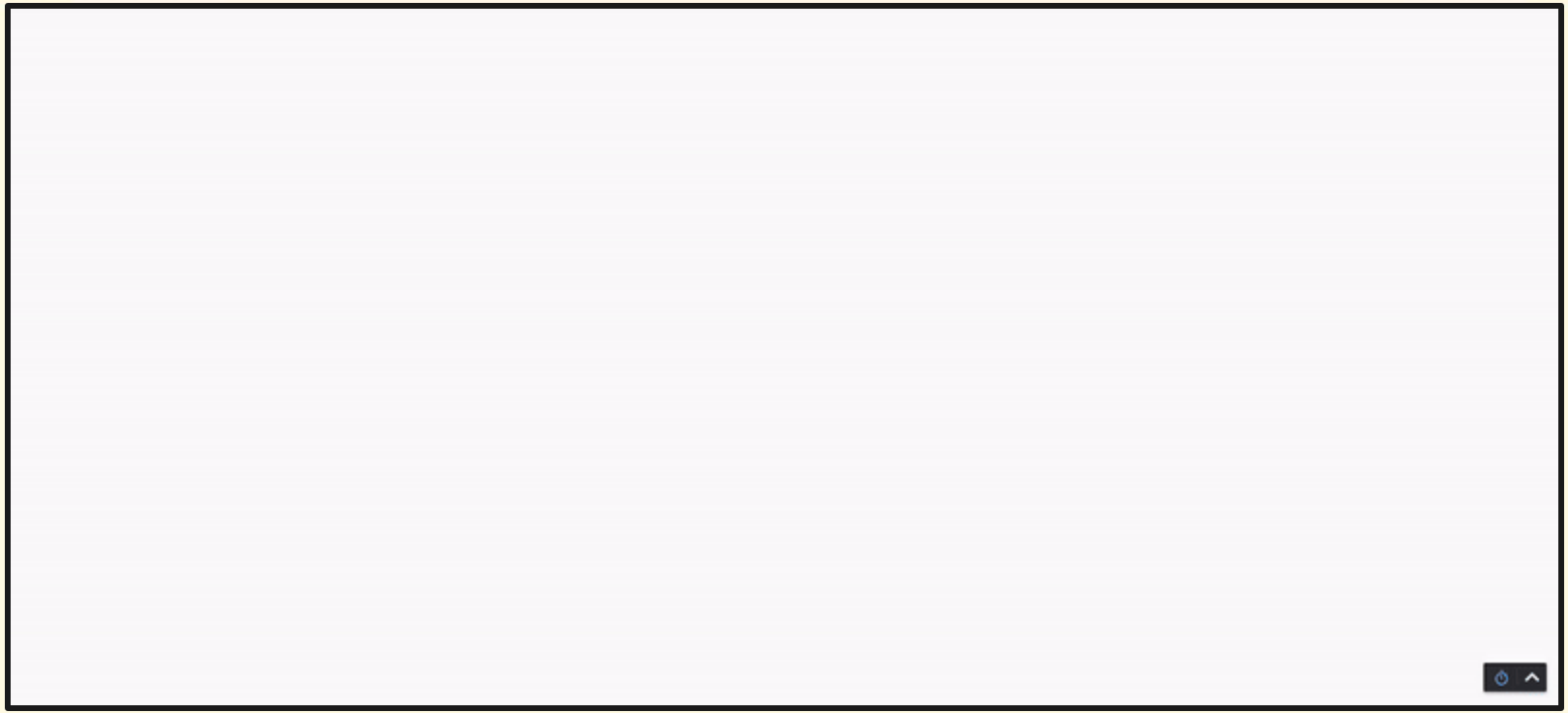
Heavyweight Gauge Component



DMC

ICC 2024

Lightweight SVG Gauge



DMC

ICC 2024

Advanced Techniques

Fundamentals

Reduce amount of data and calculation

Reduce layout recalculations

Reduce initial load actions

Reduce heavyweight components

Advanced techniques

Analyze loading

Hide loading

DMC

ICC 2024

Advanced Techniques: When to Use

Analyze Loading with DevTools

- You want to measure different variants against each other.
- You can’t figure out why your screen is still slow.

Hide Loading in Ignition

- You have optimized as much as possible, but still have pop-in.
- You want to improve page *feel.*

Fundamentals

Reduce amount of data and calculation

Reduce layout recalculations

Reduce initial load actions

Reduce heavyweight components

Advanced techniques

Analyze loading

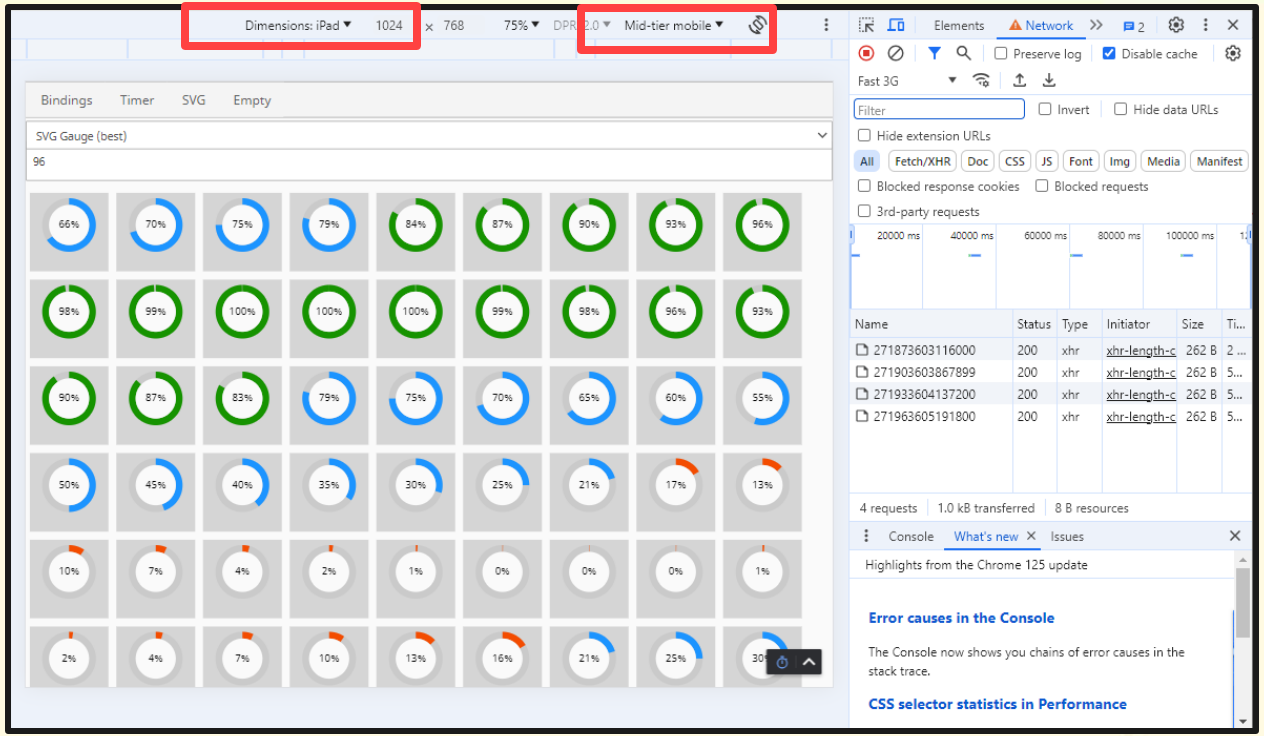
Hide loading

DMC

ICC 2024

Analyze Loading: Basic DevTools

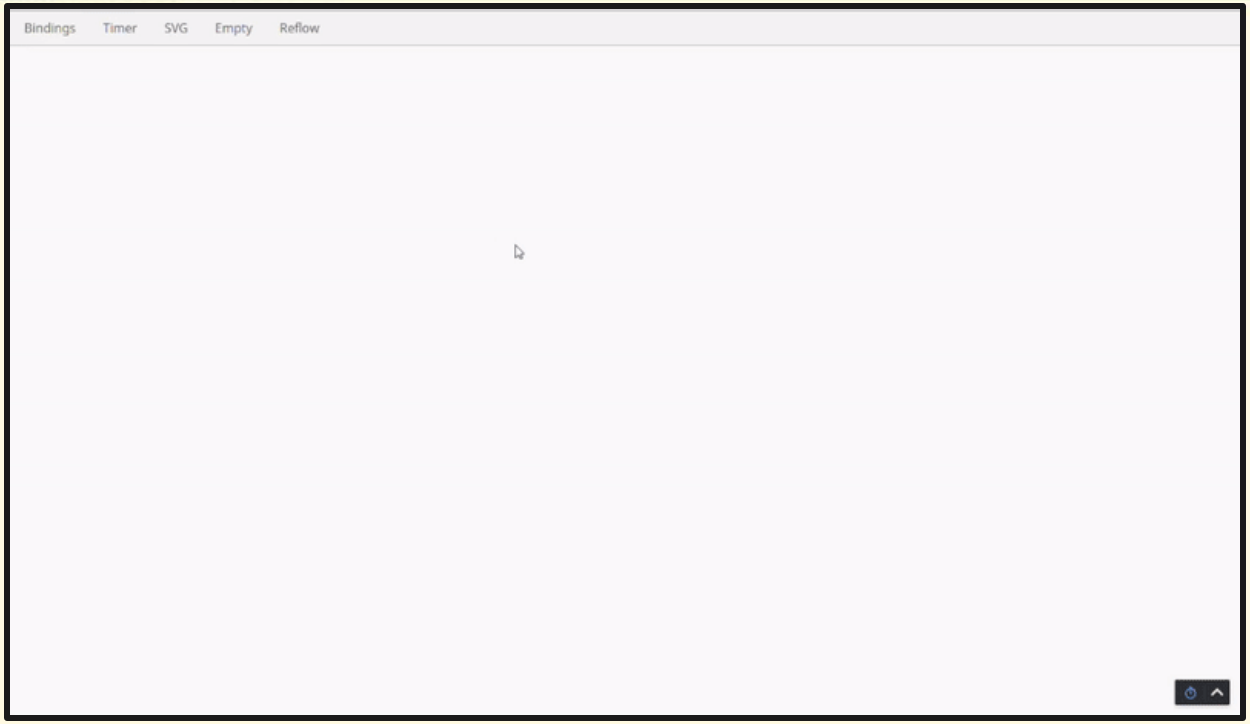
- Chrome DevTools: F12 or Ctrl+Shift+i
- Resize to test responsive design
- Throttle network speeds to exaggerate slow clients



DMC

ICC 2024

Analyze Loading: Performance Monitor



Open Chrome DevTools

↓

Switch to “Performance” tab

↓

Start recording

↓

Navigate and perform actions on your page

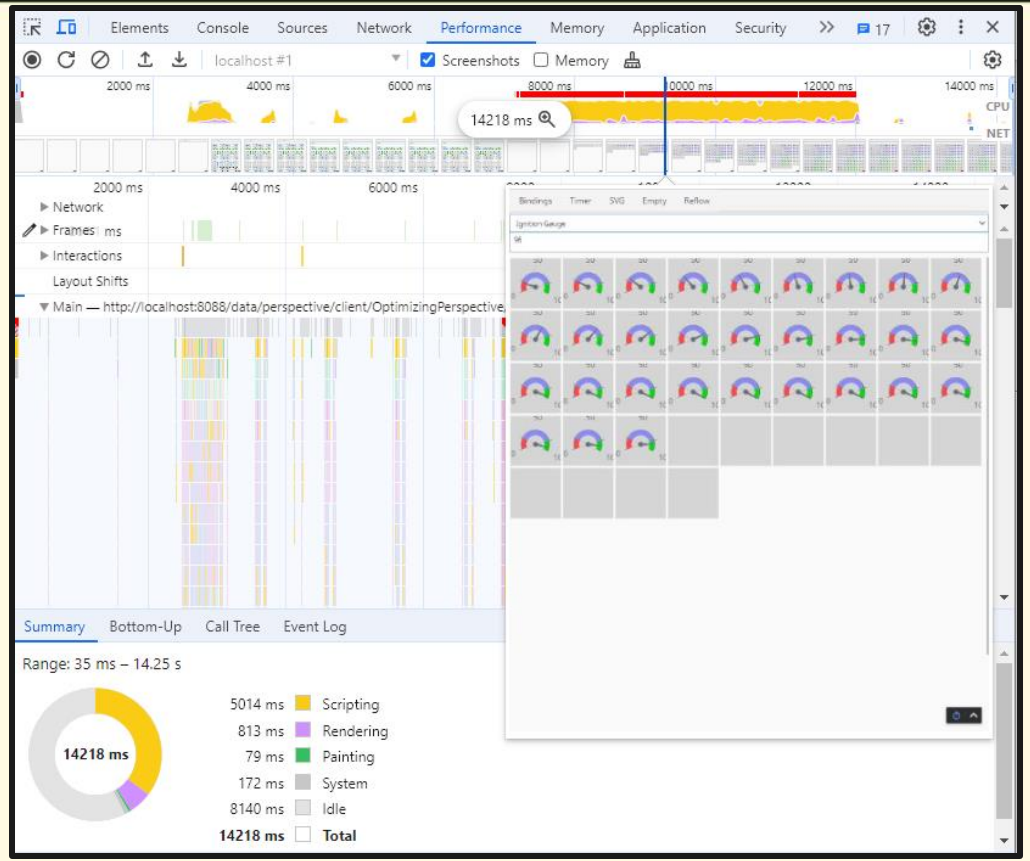
↓

Stop recording

DMC

ICC 2024

Analyze Loading: Performance Monitor



DMC

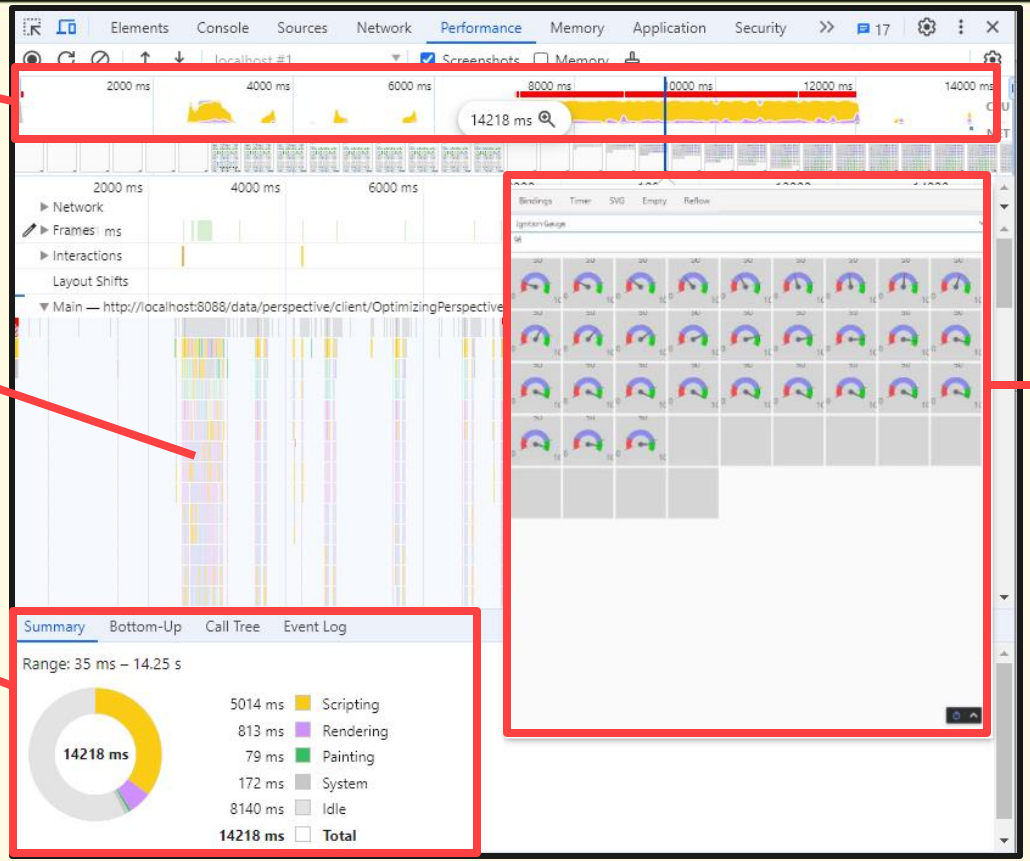
ICC 2024

Analyze Loading: Performance Monitor

Timeline
see browser resource load over time

Stack Trace
too complicated for us

Process Gauge
watch out for lots of scripting and rendering



Screenshot
hover timeline to see page snapshot

DMC

ICC 2024

Fundamentals

Reduce amount of data and calculation

Reduce layout recalculations

Reduce initial load actions

Reduce heavyweight components

Advanced techniques

Analyze loading

Hide loading

DMC

ICC 2024

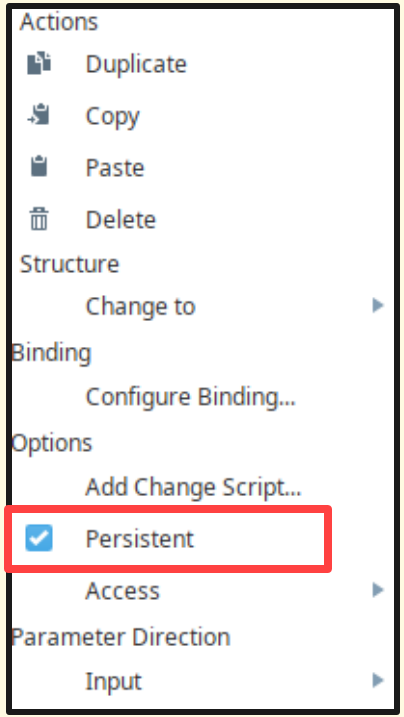
Hiding Loading

- Things don’t need to be fast, they just need to feel fast
- Some delays are unavoidable - remote tag provider bindings
- Strategies
  - Avoid overlay flickers with persistence
  - Hide with fade

DMC

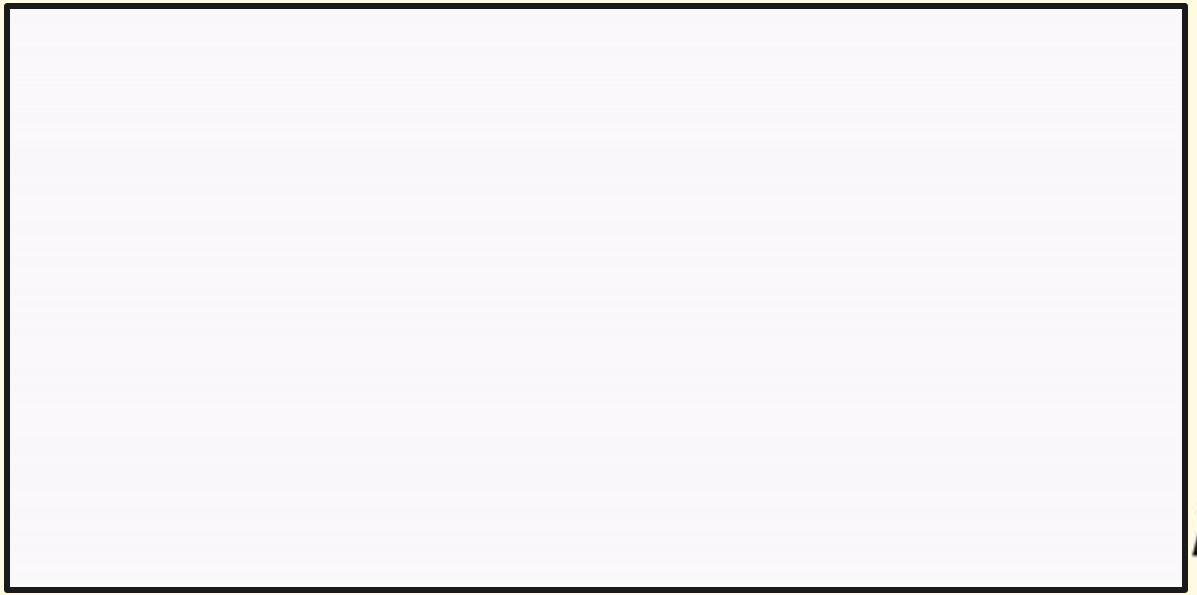
ICC 2024

Hiding Loading: Persistence

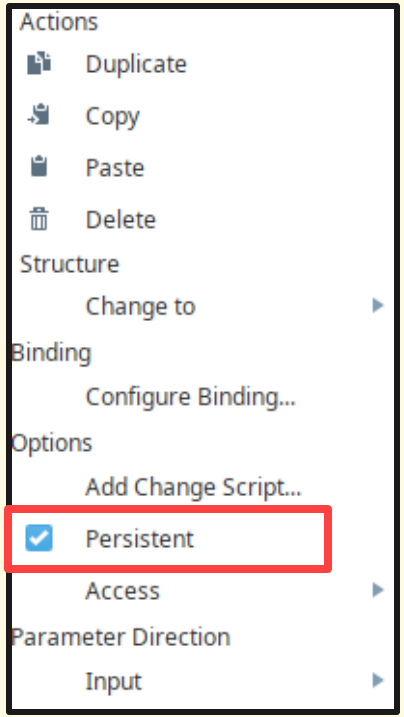


Gives the property a temporary value while waiting for the binding to load

Not Persistent – Red Overlays

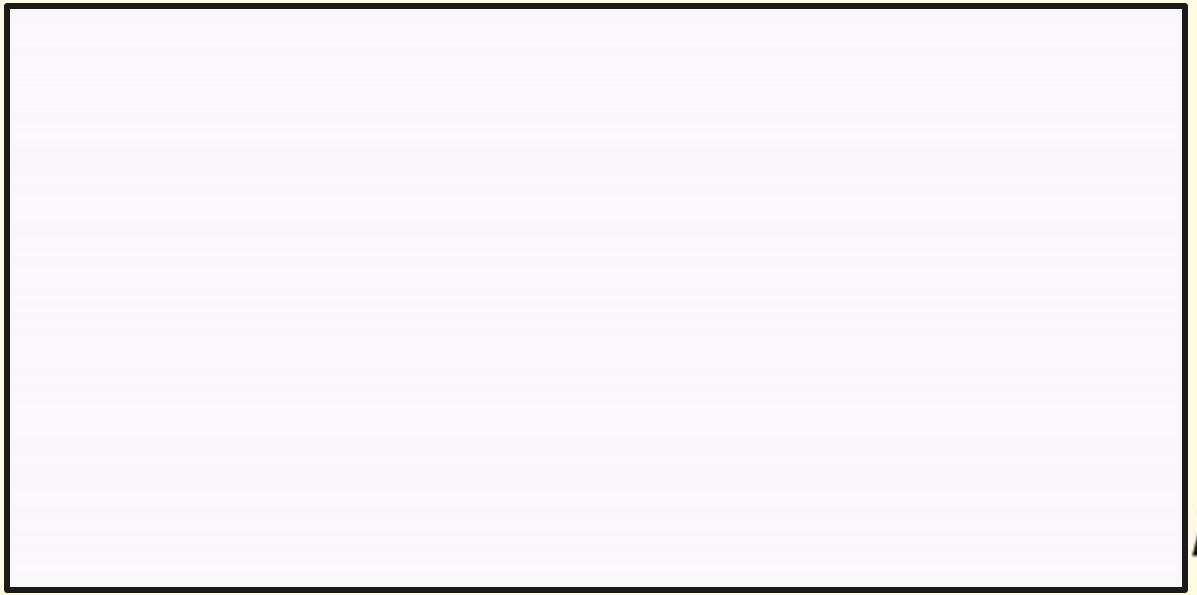


Hiding Loading: Persistence



Gives the property a temporary value while waiting for the binding to load

Persistent – Defaults to 0

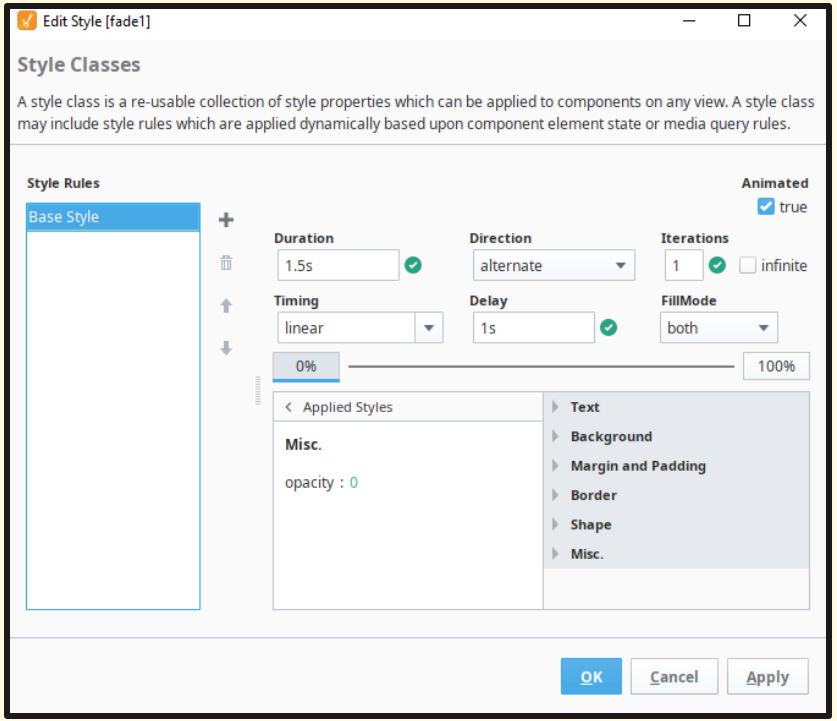


DMC

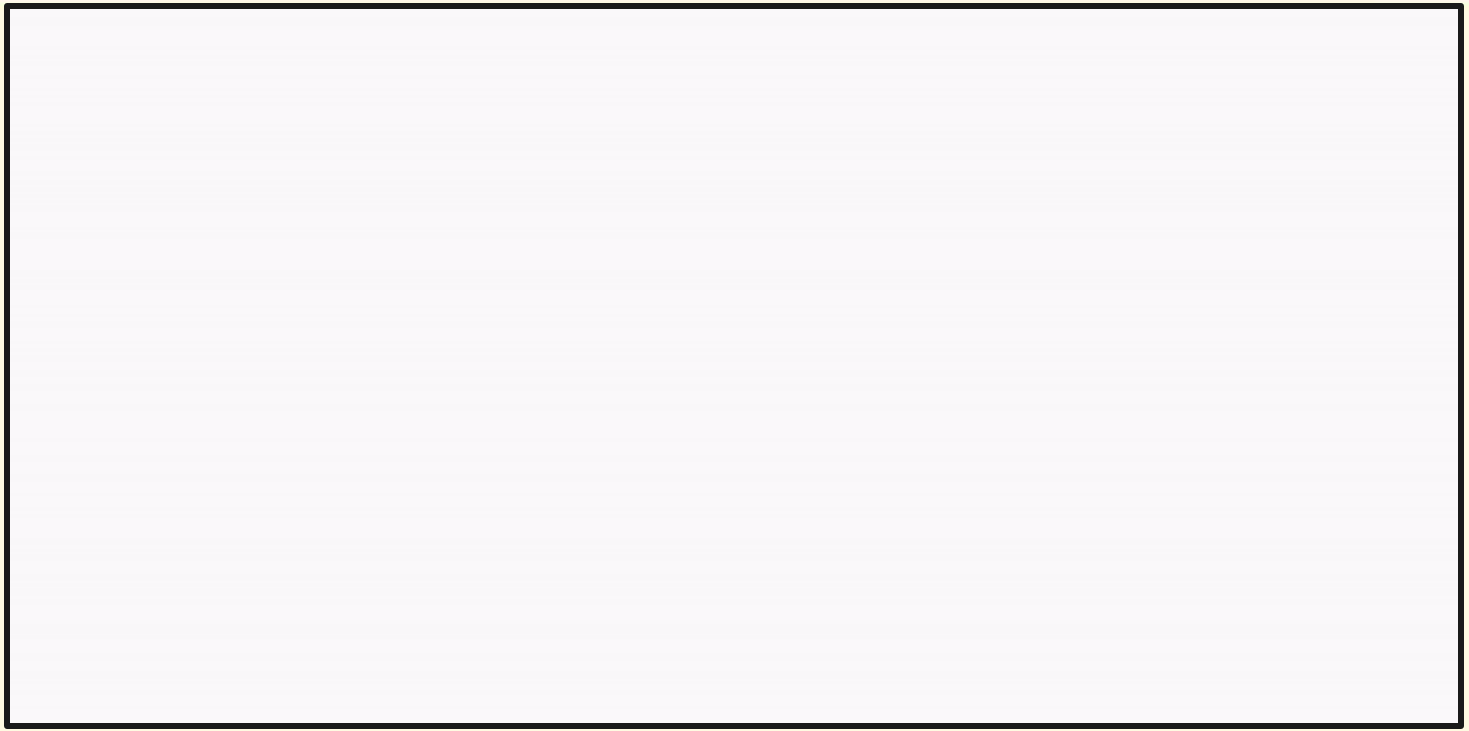
ICC 2024

Hiding Loading: Fade

- Create fade style class that animates opacity from 0 to 1
- Create multiple classes with different delay intervals



Hiding Loading: Fade



DMC

ICC 2024

Summary

Fundamentals

Reduce amount of data and calculation

Reduce layout recalculations

Reduce initial load actions

Reduce heavyweight components

Advanced techniques

Analyze loading

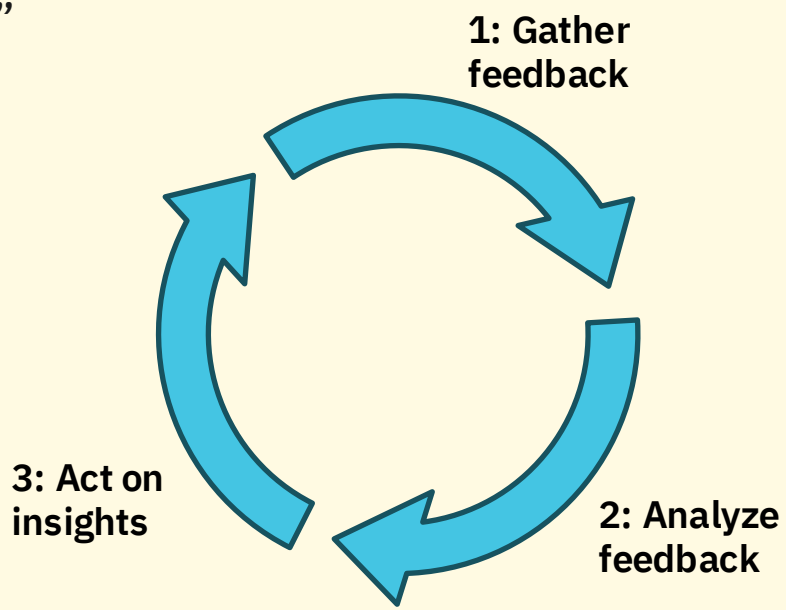
Hide loading

DMC

ICC 2024

Balance Performance vs Maintainability

- “Premature optimization is the root of all evil”
- Get user feedback, THEN consider optimizing
- Many performance boosts introduce custom or duplicate code
    - Maintainable code is always more important



Contact Us

Elizabeth Reed
elizabeth.reed@dmcinfo.com

Casimir Smith
casimir.smith@dmcinfo.com

DMC

ICC 2024

THANK YOU

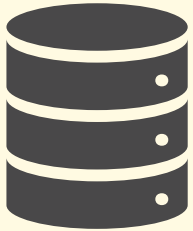
DMC

ICC 2024

Production is Often Slower Than Development

HMI/tablet is slower than engineering PC

More clients accessing server at one time



More data to query against

DMC

ICC 2024