Debugging in Xcode 11

Chris Miles, Xcode Engineering Manager
Han Ming Ong, Xcode UI Engineer
Sebastian Fischer, Xcode UI Engineer
Device Conditions and Environment Overrides
Device Conditions and Environment Overrides
Device Conditions and Environment Overrides
Device Conditions and Environment Overrides
Device Conditions and Environment Overrides
Device Conditions and Environment Overrides
Device Conditions

Thermal state condition

Raises thermal state to — fair, serious, or critical

So you can reliably

• Understand app behavior under these elevated states
• Debug and verify your handling of thermal state changes

The device does not actually get warmer
Device Conditions
Network link condition

Real world profiles simulating network conditions

- High latency
- Low bandwidth
- High packet loss
- Slow DNS response

So you can reliably

- Understand and debug app behavior under various network conditions
Environment Overrides

Interface Style
Environment Overrides

Interface Style
Environment Overrides

Interface Style

Dynamic Type

Shopping List
- Apple
- Banana
- Strawberry
- Avocado
Environment Overrides

Interface Style

Dynamic Type

Shopping List
- Apple
- Banana
- Strawberry
- Avocado

Shopping List
- Apple
- Banana
- Strawberry
- Avocado
Environment Overrides

Interface Style

Dynamic Type

Shopping List
- Apple
- Banana
- Strawberry
- Avocado

Shopping List
- Apple
- Banana
- Strawberry
- Avocado

Shopping List
- Apple
- Banana
- Strawberry
- Avocado
Environment Overrides

Interface Style

Accessibility Options

Dynamic Type

- Increase Contrast
- Bold Text
- Reduce Transparency
- Reduce Motion
- On / Off Labels
- Button Shapes
- Grayscale
- Smart Invert

Shopping List
- Apple
- Banana
- Strawberry
- Avocado
Demo

Device conditions and environment overrides
# Device Conditions

### INSTALLED APPS

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Identifier</th>
</tr>
</thead>
</table>

No apps installed

### DEVICE CONDITIONS

- **Condition**: Network Link
- **Profile**: 100% packet loss
- **Details**:
  - Name: 100% Loss Scenario
  - Downlink Bandwidth: 0 Mbps
  - Downlink Latency: 0 ms
  - Downlink Packet Loss Ratio: 100%
  - Uplink Bandwidth: 0 Mbps
  - Uplink Latency: 0 ms
  - Uplink Packet Loss Ratio: 100%

- [Start]
Device Conditions

INSTALLED APPS

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Identifier</th>
</tr>
</thead>
</table>

No apps installed

DEVICE

Condition: [ ] Network Link
[ ] Thermal State

Profile: None

- Name: 100% Loss Scenario
- Downlink Bandwidth: 0 Mbps
- Downlink Latency: 0 ms
- Downlink Packet Loss Ratio: 100%
- Uplink Bandwidth: 0 Mbps
- Uplink Latency: 0 ms
- Uplink Packet Loss Ratio: 100%

Start
Device Conditions

Device-wide induced condition

Managed by Xcode

Tied to Xcode device connection

iOS status bar indicates when a condition is active

Can be stopped on device
Device Conditions

Device-wide induced condition

Managed by Xcode

Tied to Xcode device connection

iOS status bar indicates when a condition is active

Can be stopped on device
Environment Overrides

Override runtime environment of debugged app

Overrides instantly

Only affects debugged app

System settings remain unchanged

Supports devices, simulators, and previews

Available for all platforms
Debugging Live Previews

Han Ming Ong, Xcode UI Engineer
// Day always "changed" for the first item

var dayChanged = true
var newSunAltitude: SolarAltitude? = nil
func displayables(for plan: Plan) -> ([Displayable], [Displayable]) {
  let headerDisplayable = HeaderDisplayable(plan: plan)
  var displayables: [Displayable] = []
  let events = plan.events

  for i in 0..<events.count {
    // Get event
    let event = events[i]
    var dayChanged = true
    var newSunAltitude = nil
    // Get previous event if applicable
    if let previousEvent = i > 0 ? events[i-1] {
      dayChanged = Calendar.current.isDate(event.date, inSameDayAs:previousEvent.date)
      if dayChanged {
        newSunAltitude = SolarAltitude.betweenDates(previousEvent.date, event.date)
      } else {
        newSunAltitude = .suns
      }
    } else {
      newSunAltitude = .suns
    }
    // Add displayables
    if (dayChanged) {
      displayables.append(headerDisplayable)
      displayables.append(SunDisplayable(sunAltitude: newSunAltitude))
    }
  }
  return ([headerDisplayable], displayables)
}
SwiftUI Runtime Issues
SwiftUI Runtime Issues

Found when process is running

Modifying state during view update, this will cause undefined behavior.

Thread 1

0 StoredLocation.set(_:trans...}
1 StateLink.value.setter
2 DiscoverView.__preview__bo...
3 DiscoverView.body()
38 UIApplicationMain
39 main
40 start
SwiftUI Runtime Issues

Found when process is running

Process continues executing
SwiftUI Runtime Issues

Found when process is running

Process continues executing

Details in Issues Navigator
SwiftUI Runtime Issues

Found when process is running

Process continues executing

Details in Issues Navigator
SwiftUI Runtime Issues

Found when process is running

Process continues executing

Details in Issues Navigator

All platforms and run destinations
Demo
Debugging live previews

Han Ming Ong, Debugger UI Engineer
Debugging Workflow
Debugging Workflow

Use contextual menu to start debugging
Debugging Workflow

Use contextual menu to start debugging

New debug session for source changes
Debugging Workflow

Use contextual menu to start debugging

New debug session for source changes

Keep the same file in main editor
Debugging Live Previews
Debugging Live Previews

Live previews are debuggable
Debugging Live Previews

Live previews are debuggable

Workflows for previews
Debugging Live Previews

Live previews are debuggable

Workflows for previews

SwiftUI runtime issues
Debugging SwiftUI View Hierarchies

Sebastian Fischer, Xcode UI Engineer
Debugging SwiftUI View Hierarchies

Declarative API

Composition and lightweight modifiers over complex views

Value type semantics

Platform specific views at runtime

Interoperability with UIKit and AppKit
Demo

Debugging SwiftUI view hierarchies

Sebastian Fischer, Xcode UI Engineer
Debugging SwiftUI View Hierarchies

View hierarchy in navigator and canvas

Properties and modifiers in inspector

Automatic inspector properties through Swift reflection

Custom inspector properties by adopting `CustomReflectable`

Support for mixed view hierarchies
View Hierarchy Debugging Improvements

UIWindowScene support

Inspectors
• Trait collections
• Named images
• Symbol images
• Named colors
• Improved constraint details
More Information

developer.apple.com/wwdc19/412

---

<table>
<thead>
<tr>
<th>Designing for Adverse Network and Temperature Conditions</th>
<th>Friday, 4:20</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLDB: Beyond &quot;po&quot;</td>
<td>WWDC 2019</td>
</tr>
</tbody>
</table>