

#WWDC18

# Advanced Debugging with Xcode and LLDB

Session 412

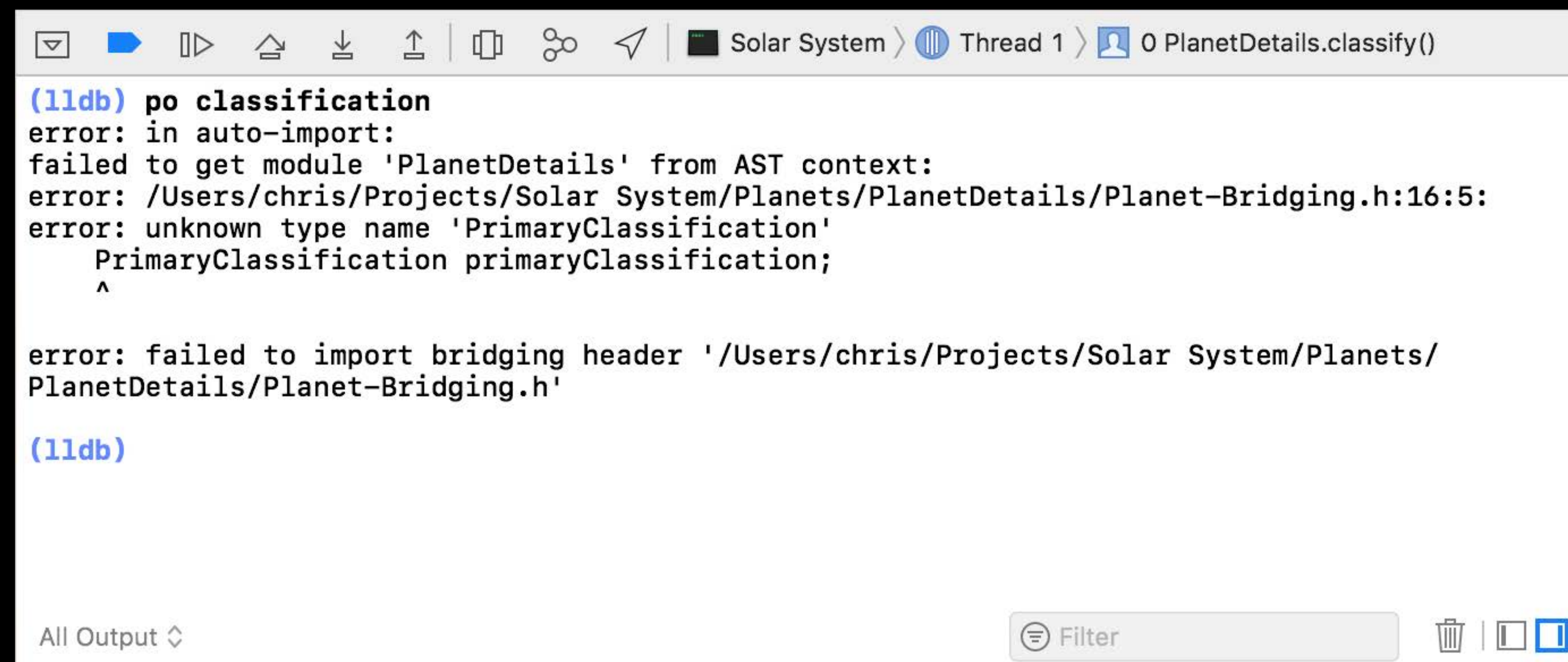
Chris Miles, Xcode Engineering Manager  
Sebastian Fischer, Xcode UI Engineer

# Swift Debugging Reliability

Chris Miles, Xcode Engineering Manager

**“Failed to get module from AST context”**

# "Failed to get module from AST context"



The screenshot shows the Xcode 9 console output for a thread named 'Thread 1' at the location '0 PlanetDetails.classify()'. The output contains the following text:

```
(lldb) po classification
error: in auto-import:
failed to get module 'PlanetDetails' from AST context:
error: /Users/chris/Projects/Solar System/Planets/PlanetDetails/Planet-Bridging.h:16:5:
error: unknown type name 'PrimaryClassification'
    PrimaryClassification primaryClassification;
    ^
error: failed to import bridging header '/Users/chris/Projects/Solar System/Planets/
PlanetDetails/Planet-Bridging.h'

(lldb)
```

At the bottom of the console, there is a filter input field containing the text 'Filter' and some navigation icons.

Xcode 9



# "Failed to get module from AST context"

NEW

```
Solar System > Thread 1 > 0 PlanetDetails.classify()
(11db) po classification
error: in auto-import:
failed to get module 'PlanetDetails' from AST context:
error: /Users/chris/Projects/Solar System/Planets/PlanetDetails/Planet-Bridging.h:16:5:
error: unknown type name 'PrimaryClassification'
    PrimaryClassification primaryClassification;
    ^
error: failed to import bridging header '/Users/chris/Projects/Solar System/Planets/
PlanetDetails/Planet-Bridging.h'
(11db)
```

Xcode 9

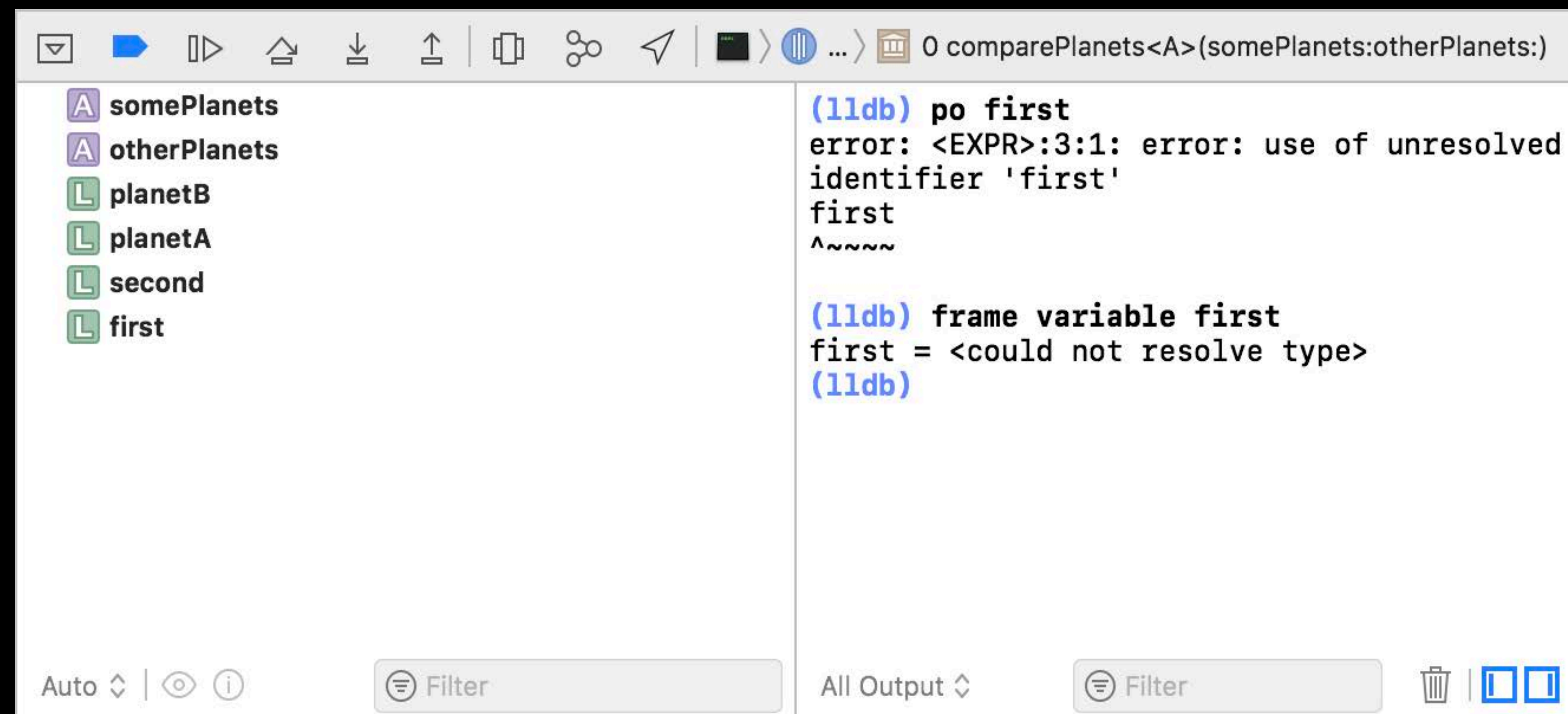
```
Solar System > Thread 1 > 0 PlanetDetails.classify()
(11db) po classification
note: Swift compiler options for Solar System conflict with options found in other modules;
      Switching to a new expression evaluator for Solar System, old $R variables are lost.
- PlanetClassification
  - primaryClassification : PrimaryClassification
    - rawValue : 2
(11db)
```

Xcode 10

# Swift Type Resolution



# Swift Type Resolution

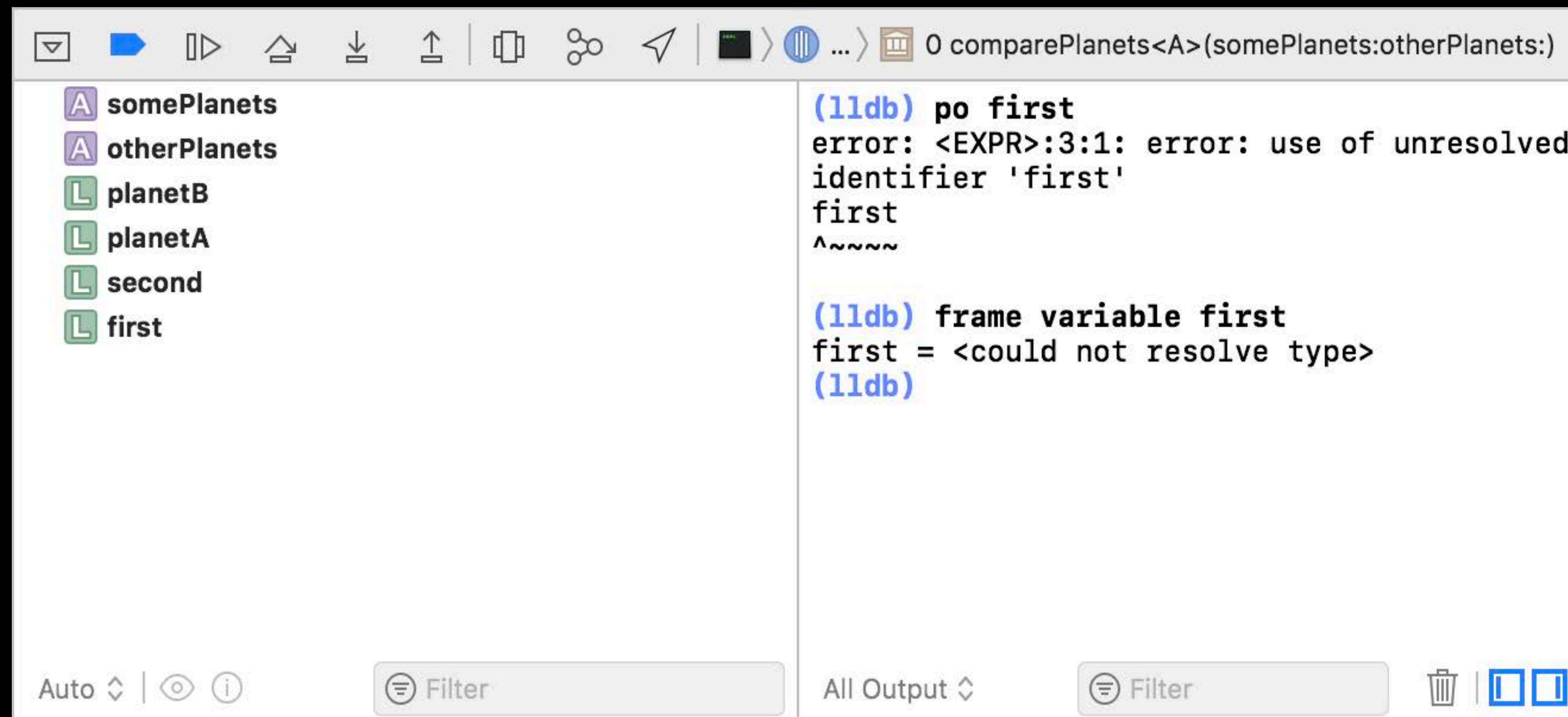


Xcode 9



# Swift Type Resolution

NEW

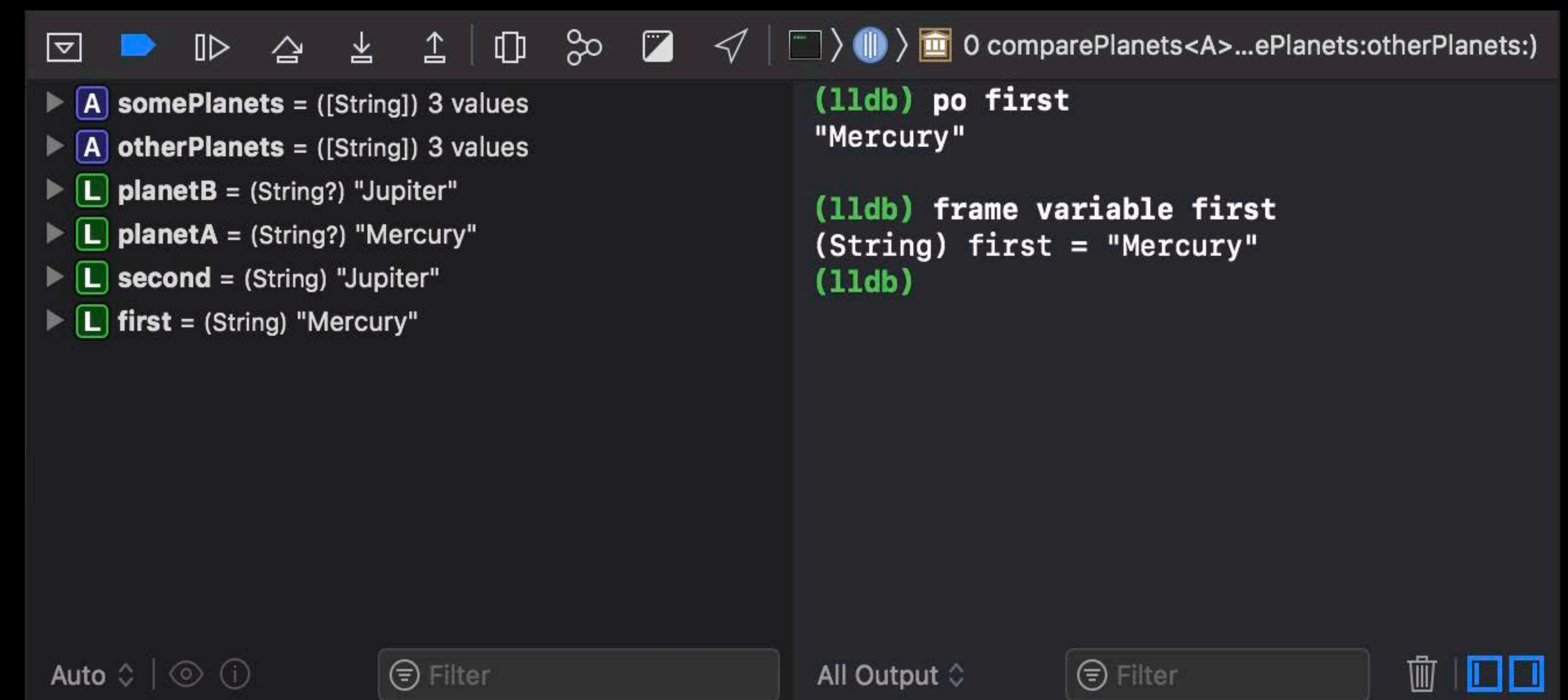


The screenshot shows the Xcode 9 interface. On the left, a variable declaration list includes 'somePlanets', 'otherPlanets', 'planetB', 'planetA', 'second', and 'first'. The main editor displays the following Swift code and REPL output:

```
(lldb) po first
error: <EXPR>:3:1: error: use of unresolved
identifier 'first'
first
^~~~~

(lldb) frame variable first
first = <could not resolve type>
(lldb)
```

Xcode 9



The screenshot shows the Xcode 10 interface. On the left, the variable declaration list is expanded to show the types for each variable: 'somePlanets' and 'otherPlanets' are arrays of strings, 'planetB' is an optional string, 'planetA' is a string, 'second' is a string, and 'first' is a string. The main editor displays the following Swift code and REPL output:

```
(lldb) po first
"Mercury"

(lldb) frame variable first
(String) first = "Mercury"
(lldb)
```

Xcode 10



# Advanced Debugging Tips and Tricks

Chris Miles, Xcode Engineering Manager

# *Demo*

Chris Miles, Xcode Engineering Manager

# Advanced Debugging Tips and Tricks

Configure behaviors to dedicate a tab for debugging

LLDB expressions can modify program state

Use auto-continuing breakpoints with debugger commands to inject code live

Create dependent breakpoints using `breakpoint set --one-shot true`

`po $arg1` (`$arg2`, etc) in assembly frames to print function arguments

Skip lines of code by dragging Instruction Pointer or `thread jump --by 1`

Pause when variables are modified by using watchpoints



# Advanced Debugging Tips and Tricks

Evaluate Obj-C code in Swift frames with `expression -l objc -0 -- <expr>`

Flush view changes to the screen using `expression CATransaction.flush()`

Add custom LLDB commands using aliases and scripts. Alias examples:

```
command alias poc expression -l objc -0 --
```

```
command alias 🍻 expression -l objc -- (void)[CATransaction flush]
```

The nudge LLDB script will be available from the session website  
<https://developer.apple.com/wwdc18/412>

# LLDB Print Commands

Command	Alias For	Steps To Evaluate
<code>po &lt;expression&gt;</code>	<code>expression --object-description -- &lt;expression&gt;</code>	<ol style="list-style-type: none"><li>1. Expression: evaluate <i>&lt;expression&gt;</i></li><li>2. Expression: debug description</li></ol>

# LLDB Print Commands

Command	Alias For	Steps To Evaluate
<code>po &lt;expression&gt;</code>	<code>expression --object-description -- &lt;expression&gt;</code>	<ol style="list-style-type: none"><li>1. Expression: evaluate <i>&lt;expression&gt;</i></li><li>2. Expression: debug description</li></ol>
<code>p &lt;expression&gt;</code>	<code>expression -- &lt;expression&gt;</code>	<ol style="list-style-type: none"><li>1. Expression: evaluate <i>&lt;expression&gt;</i></li><li>2. Outputs LLDB-formatted description</li></ol>



# LLDB Print Commands

Command	Alias For	Steps To Evaluate
<code>po &lt;expression&gt;</code>	<code>expression --object-description -- &lt;expression&gt;</code>	<ol style="list-style-type: none"><li>1. Expression: evaluate <i>&lt;expression&gt;</i></li><li>2. Expression: debug description</li></ol>
<code>p &lt;expression&gt;</code>	<code>expression -- &lt;expression&gt;</code>	<ol style="list-style-type: none"><li>1. Expression: evaluate <i>&lt;expression&gt;</i></li><li>2. Outputs LLDB-formatted description</li></ol>
<code>frame variable &lt;name&gt;</code>		<ol style="list-style-type: none"><li>1. Reads value of <i>&lt;name&gt;</i> from memory</li><li>2. Outputs LLDB-formatted description</li></ol>

# Advanced View Debugging

Sebastian Fischer, Xcode UI Engineer

# *Demo*

Sebastian Fischer, Xcode UI Engineer



# View Debugging Tips

Reveal in Debug Navigator

View clipped content

Auto Layout debugging

Access object pointers (copy casted expressions)

Creation backtraces in the inspector

Debug description in the inspector

⌘-click-through for selection

# View Debugging Tips

Reveal in Debug Navigator

View clipped content

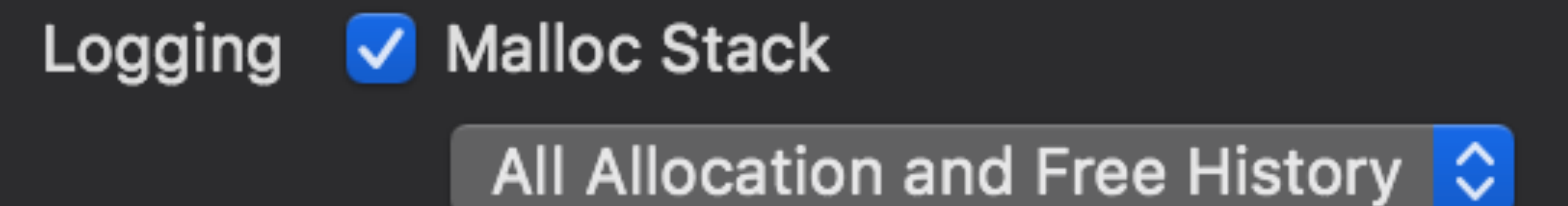
Auto Layout debugging

Access object pointers (copy casted expressions)

Creation backtraces in the inspector 

Debug description in the inspector

⌘-click-through for selection



# Debugging Dark Mode

Appearance overrides

Capturing active Mac apps

Named colors and NSAppearance details in inspector

---

Introducing Dark Mode

WWDC 2018

---

Advanced Dark Mode

WWDC 2018

---

# Debugging Dark Mode

NEW

Appearance overrides

Capturing active Mac apps

Named colors and NSAppearance details in inspector

---

Introducing Dark Mode

WWDC 2018

---

Advanced Dark Mode

WWDC 2018

---



# More Information

<https://developer.apple.com/wwdc18/412>

---

Profiling and Debugging Lab

Technology Lab 10

Friday 9:00AM

---

iOS Memory Deep Dive

Hall 1

Friday 2:00PM

---

 **WWDC18**