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”
"Failed to get module from AST context"

```
(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'
   ^
error: failed to import bridging header '/Users/chris/Projects/Solar System/Planets/
   PlanetDetails/Planet-Bridging.h'
(11db)
```

Xcode 9

```
(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

Xcode 9

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 -O -- <expr>"

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

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

```bash
command alias poc expression -l objc -O --
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

<table>
<thead>
<tr>
<th>Command</th>
<th>Alias For</th>
<th>Steps To Evaluate</th>
</tr>
</thead>
</table>
| `po <expression>` | `expression --object-description -- <expression>` | 1. Expression: evaluate `<expression>`  
2. Expression: debug description |
# LLDB Print Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Alias For</th>
<th>Steps To Evaluate</th>
</tr>
</thead>
</table>
| `po <expression>` | `expression --object-description -- <expression>` | 1. Expression: evaluate `<expression>`  
2. Expression: debug description |
| `p <expression>`     | `expression -- <expression>`                  | 1. Expression: evaluate `<expression>`  
2. Outputs LLDB-formatted description |
## LLDB Print Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Alias For</th>
<th>Steps To Evaluate</th>
</tr>
</thead>
<tbody>
<tr>
<td>po &lt;expression&gt;</td>
<td>expression --object-description -- &lt;expression&gt;</td>
<td>1. Expression: evaluate <code>&lt;expression&gt;</code>&lt;br&gt;2. Expression: debug description</td>
</tr>
<tr>
<td>p &lt;expression&gt;</td>
<td>expression -- &lt;expression&gt;</td>
<td>1. Expression: evaluate <code>&lt;expression&gt;</code>&lt;br&gt;2. Outputs LLDB-formatted description</td>
</tr>
<tr>
<td>frame variable &lt;name&gt;</td>
<td></td>
<td>1. Reads value of <code>&lt;name&gt;</code> from memory&lt;br&gt;2. Outputs LLDB-formatted description</td>
</tr>
</tbody>
</table>
Advanced View Debugging
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

Advanced Dark Mode
Debugging Dark Mode

Appearance overrides

Capturing active Mac apps

Named colors and NSAppearance details in inspector

Introducing Dark Mode

Advanced Dark Mode

WWDC 2018
<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profiling and Debugging Lab</td>
<td>Technology Lab 10</td>
<td>Friday 9:00AM</td>
</tr>
<tr>
<td>iOS Memory Deep Dive</td>
<td>Hall 1</td>
<td>Friday 2:00PM</td>
</tr>
</tbody>
</table>

More Information

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