

# Debugging with Xcode

Joar Wingfors, [joar@joar.com](mailto:joar@joar.com)

(Cell Phones ► Silent)

Whirlwind!

# Introduction

- `printf()` / `NSLog()` vs. Breakpoints
- Familiarity & Habit
- Performance
  - Launching
  - Iterating

# Run vs. Debug

- Run, then Debug
  - Activate / Deactivate breakpoints
- Auto-attach on crash
- Go?

# “Smart” Breakpoints

- Properties
  - Actions, Auto-Continue, Condition & Hit Count
- Actions
  - Command, Log, Sound, Script
- Templates
  - Log stack trace and auto-continue
- Project vs. Global

- Go (Debug) ⌘⇧↩
- Run ⌘⇧R
- Debug ⌘⇧Y
- Start with Performance Tool ▶
- Stop ⌘⇧↩
- Attach to Process ▶

---

- Debugger ⌘⇧Y
- Mini Debugger ⌘⇧↩
- Console ⌘⇧R

---

- Show ▶
- Debugger Display ▶
- Variables View ▶

---

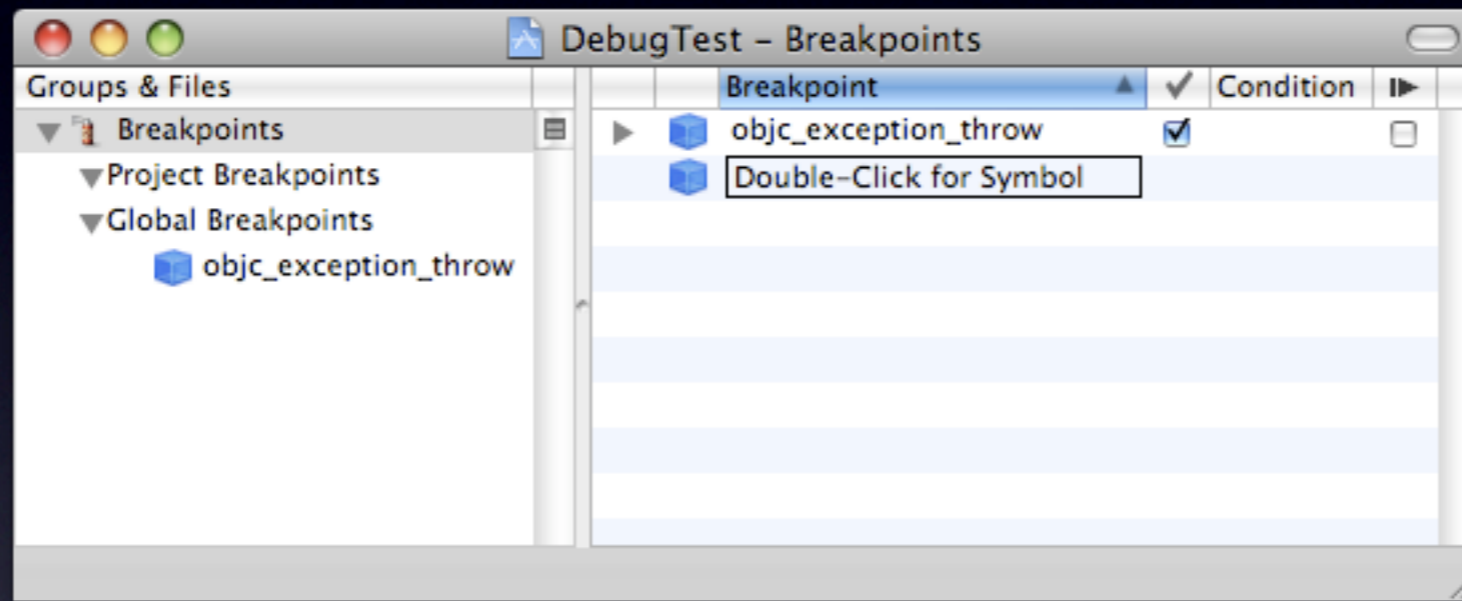
- Deactivate Breakpoints ⌘⇧\
- Manage Breakpoints ▶
- Fix
- Pause ⌘⇧P
- Step Into ⌘⇧I
- Step Over ⌘⇧O
- Step Out ⌘⇧T

---

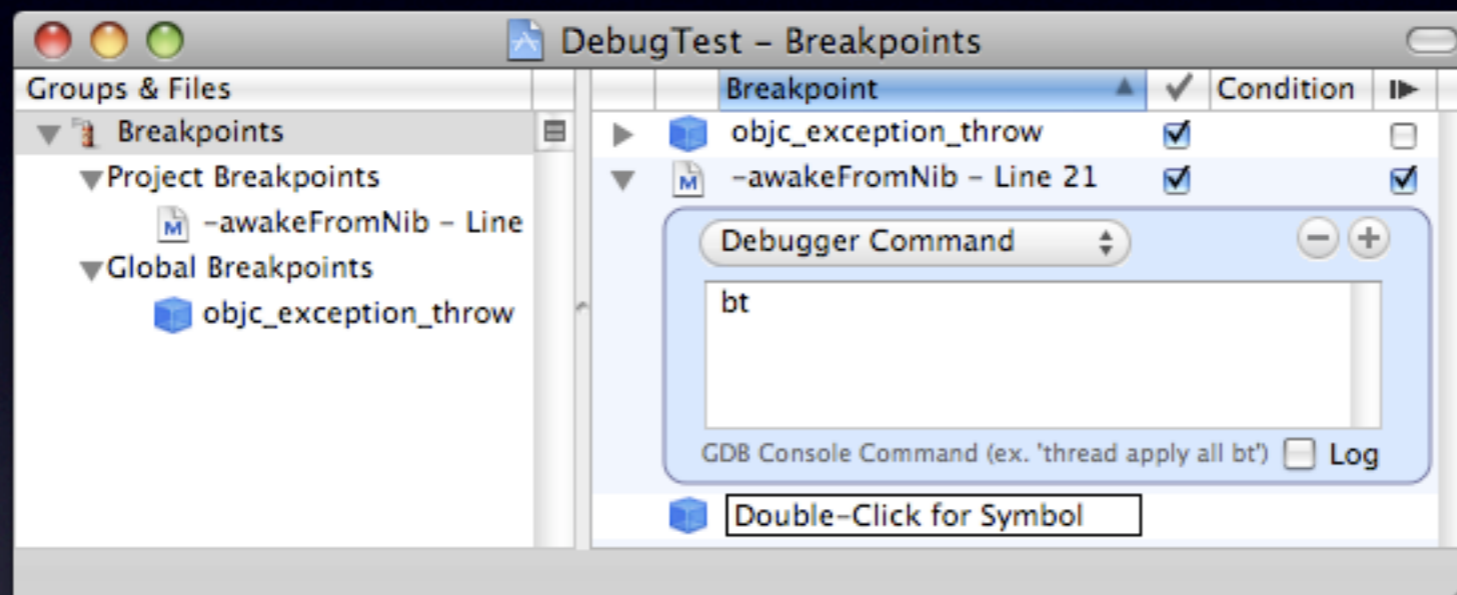
- Sync With Debugger
- ✓ Stop on Debugger()/DebugStr()
- Enable Guard Malloc

- Go (Debug) ⌘⇧↩
- Run ⌘⇧R
- Debug ⌘⇧Y
- Start with Performance Tool ▶
- Stop ⌘⇧↩
- Attach to Process ▶
- Debugger ⌘⇧Y
- Mini Debugger ⌘⇧↩
- Console ⌘⇧R
- Show ▶
- Debugger Display ▶
- Variables View ▶
- Deactivate Breakpoints ⌘⇧\
- Manage Breakpoints ▶
- Fix
- Pause ⌘⇧P
- Step Into ⌘⇧I
- Step Over ⌘⇧O
- Step Out ⌘⇧T
- Sync With Debugger
- ✓ Stop on Debugger()/DebugStr()
- Enable Guard Malloc

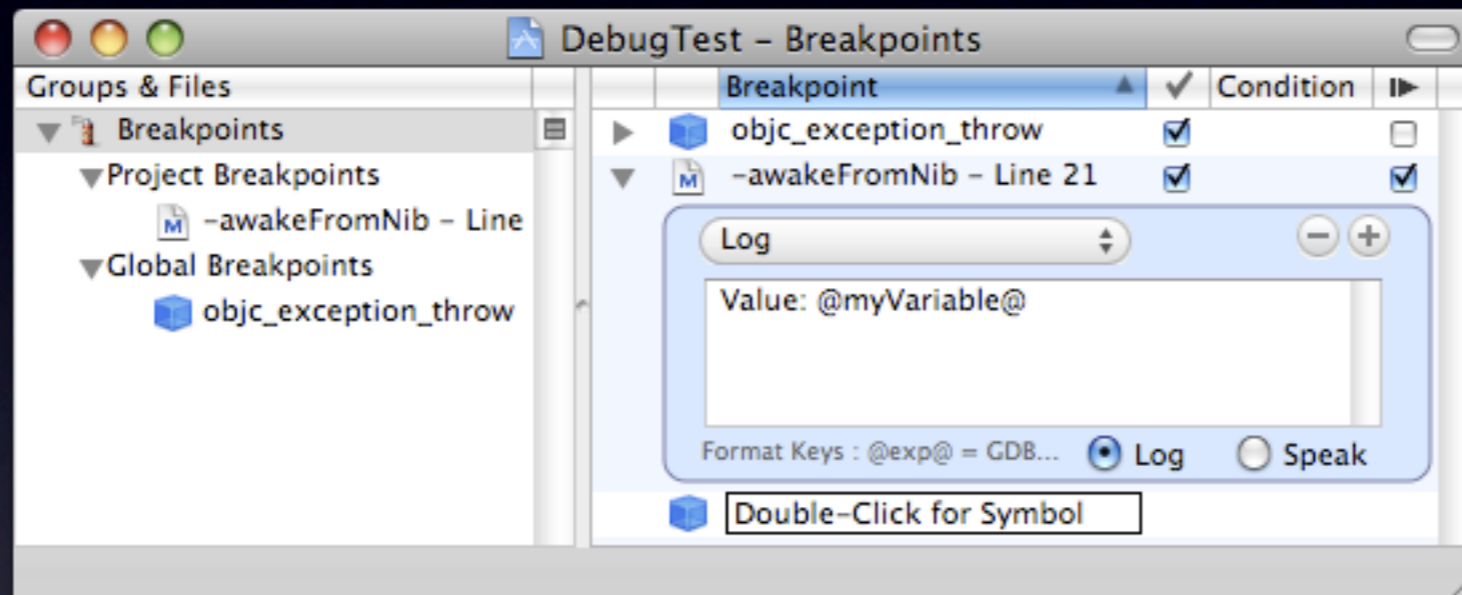
- Breakpoints ⌘⇧B
- Expressions...
- Global Variables...
- Memory Browser...
- Shared Libraries...



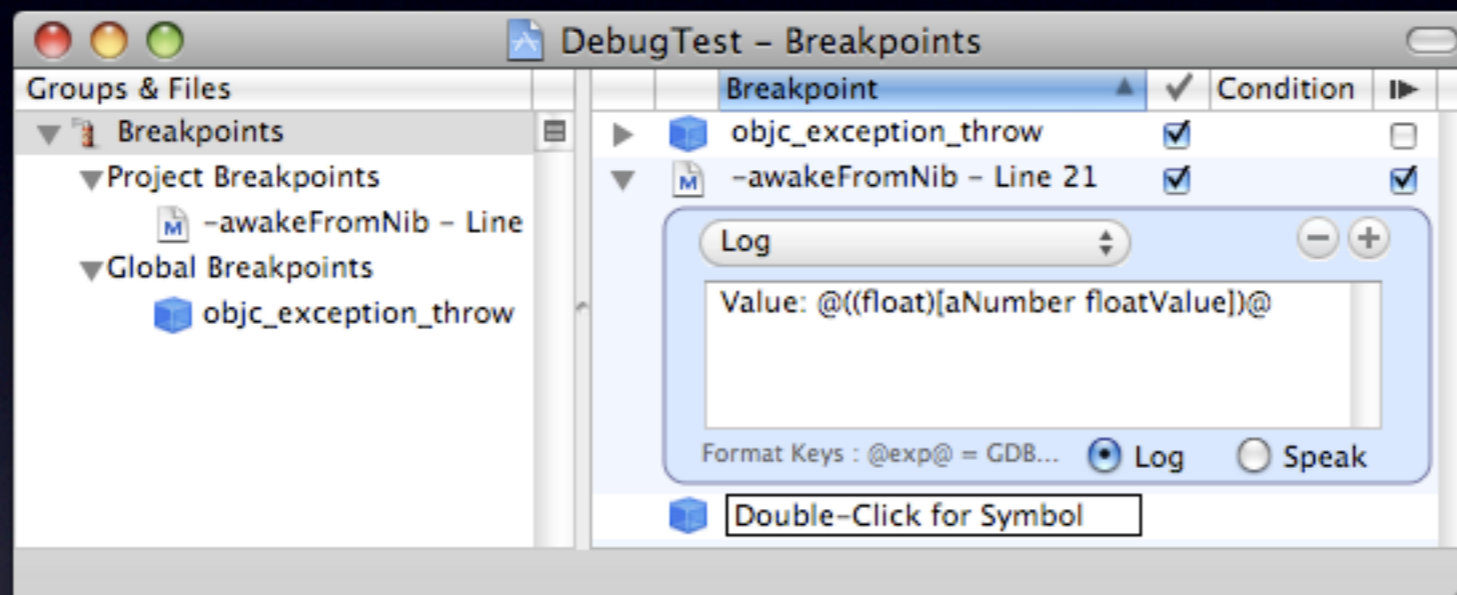
## Global Breakpoints



Log stack trace and auto-continue



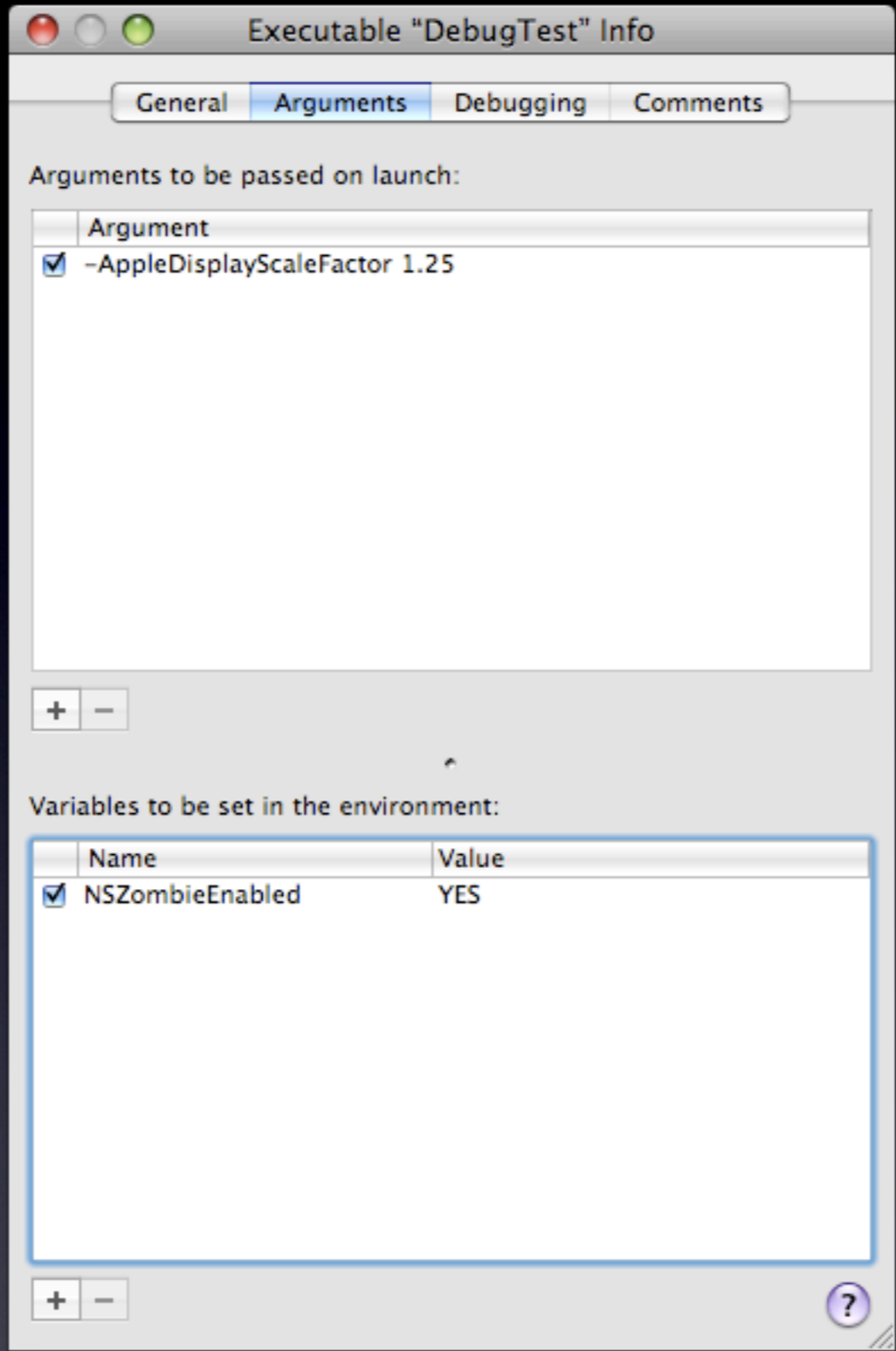
Print variables



Print the result of expressions

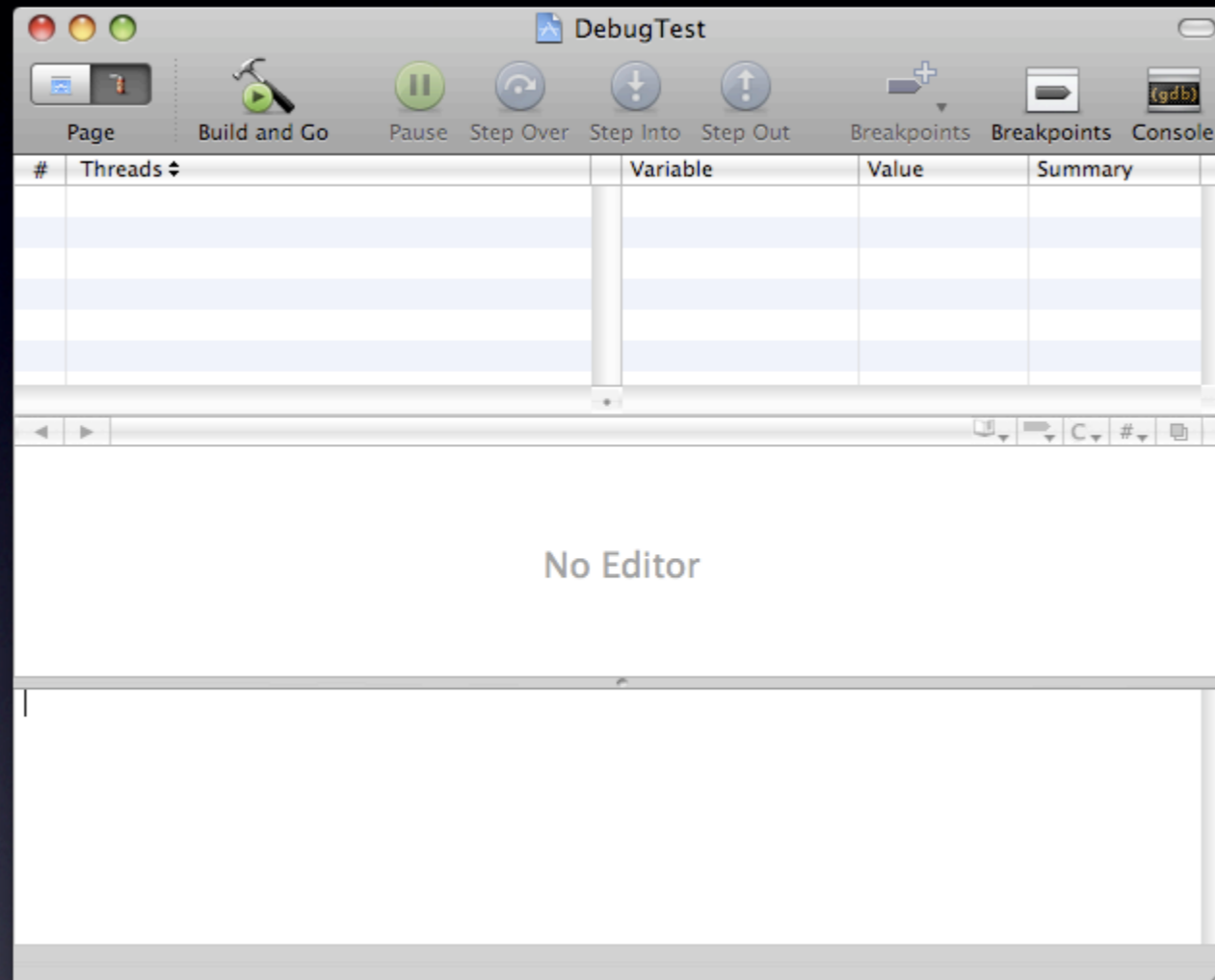
# Executables

- Target vs. Custom
- Arguments & Environment
  - NSZombieEnabled YES
  - -AppleDisplayScaleFactor 1.25
- Auto-Attach preference

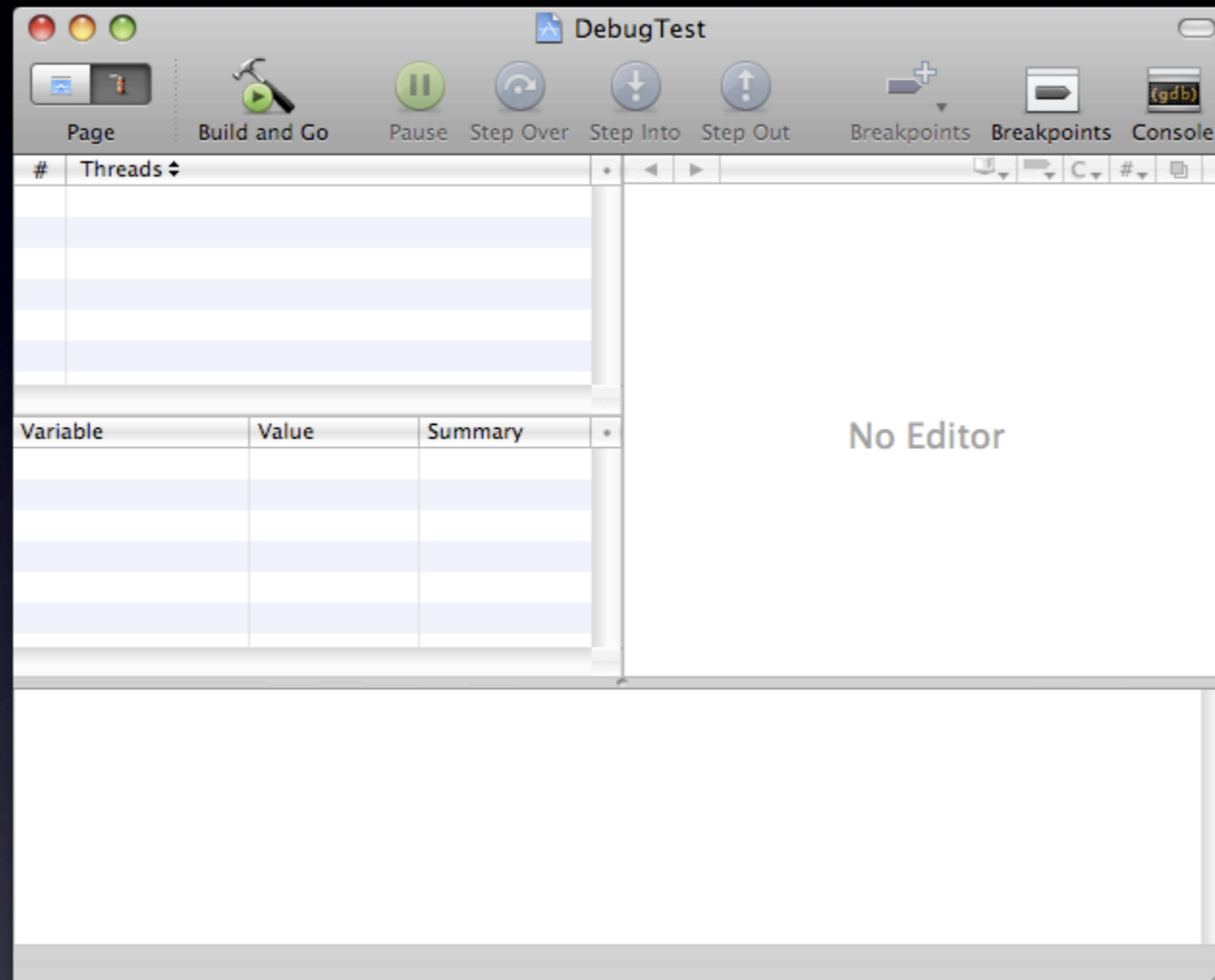


# Debugger UI

- Vertical vs. Horizontal
- Mini Debugger, “HUD”
- Debugger Strip
- Debugger Datatips
- Console



Horizontal Layout



Vertical Layout

- Go (Debug) ⌘⇧↩
- Run ⌘⇧R
- Debug ⌘⇧Y
- Start with Performance Tool ▶
- Stop ⌘⇧↩
- Attach to Process ▶
- Debugger ⌘⇧Y
- Mini Debugger ^⇧↖
- Console ⌘⇧R
- Show ▶
- Debugger Display** ▶
- Variables View ▶
- Deactivate Breakpoints ^⇧\
- Manage Breakpoints ▶
- Fix
- Continue ⌘⇧P
- Step Into ⌘⇧I
- Step Over ⌘⇧O
- Step Out ⌘⇧T
- Sync With Debugger
- ✓ Stop on Debugger()/DebugStr()
- Enable Guard Malloc

- Horizontal Layout
- ✓ Vertical Layout
- ✓ **Datatips**
- ✓ Step Controls
- ✓ Source Only
- Source and Disassembly
- Disassembly Only

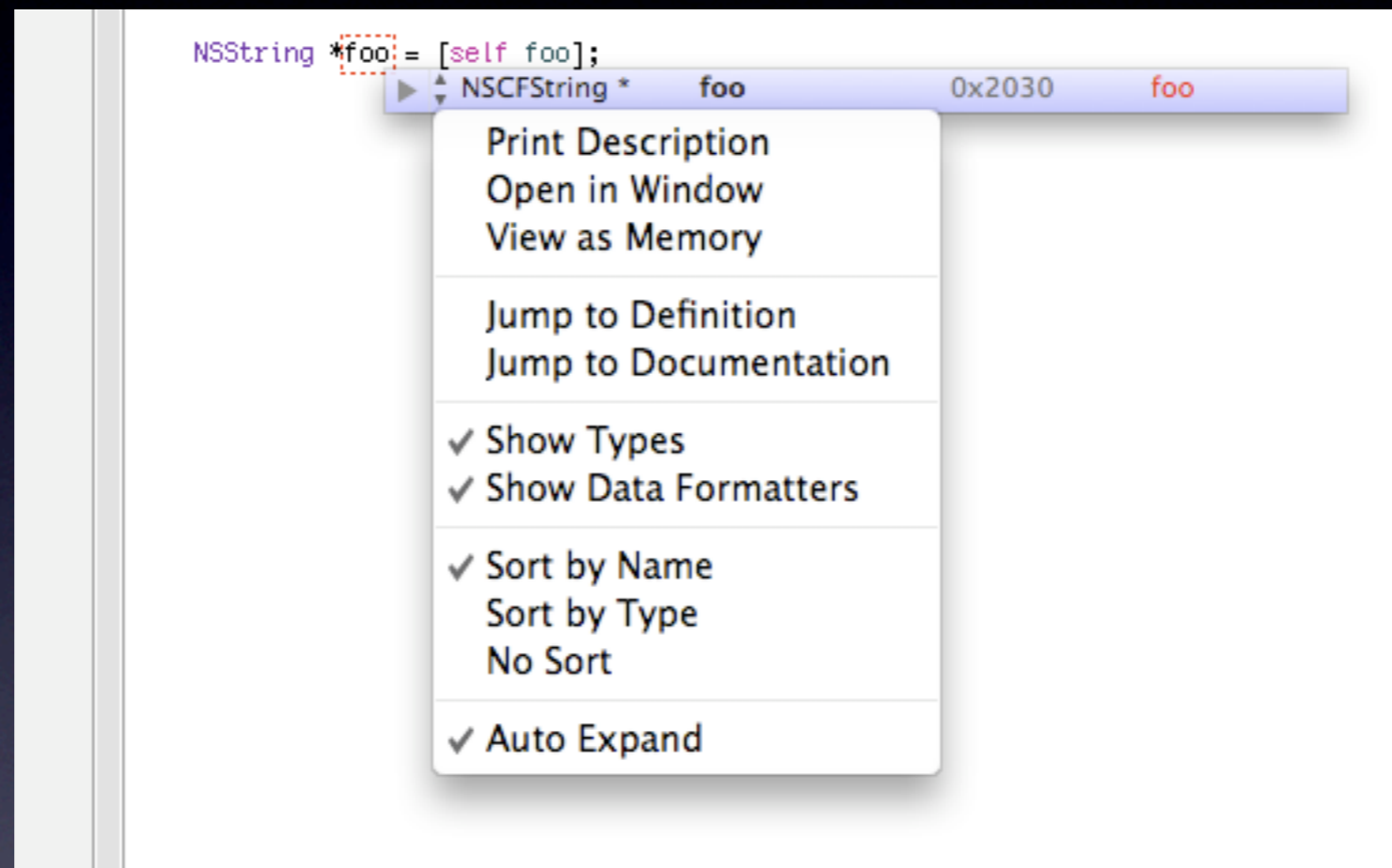
```
NSString *foo = [self foo];
```



The datatip shows the following information:

NSString *	foo	0x2030	foo
------------	-----	--------	-----

# Datatips



## Datatips Menu

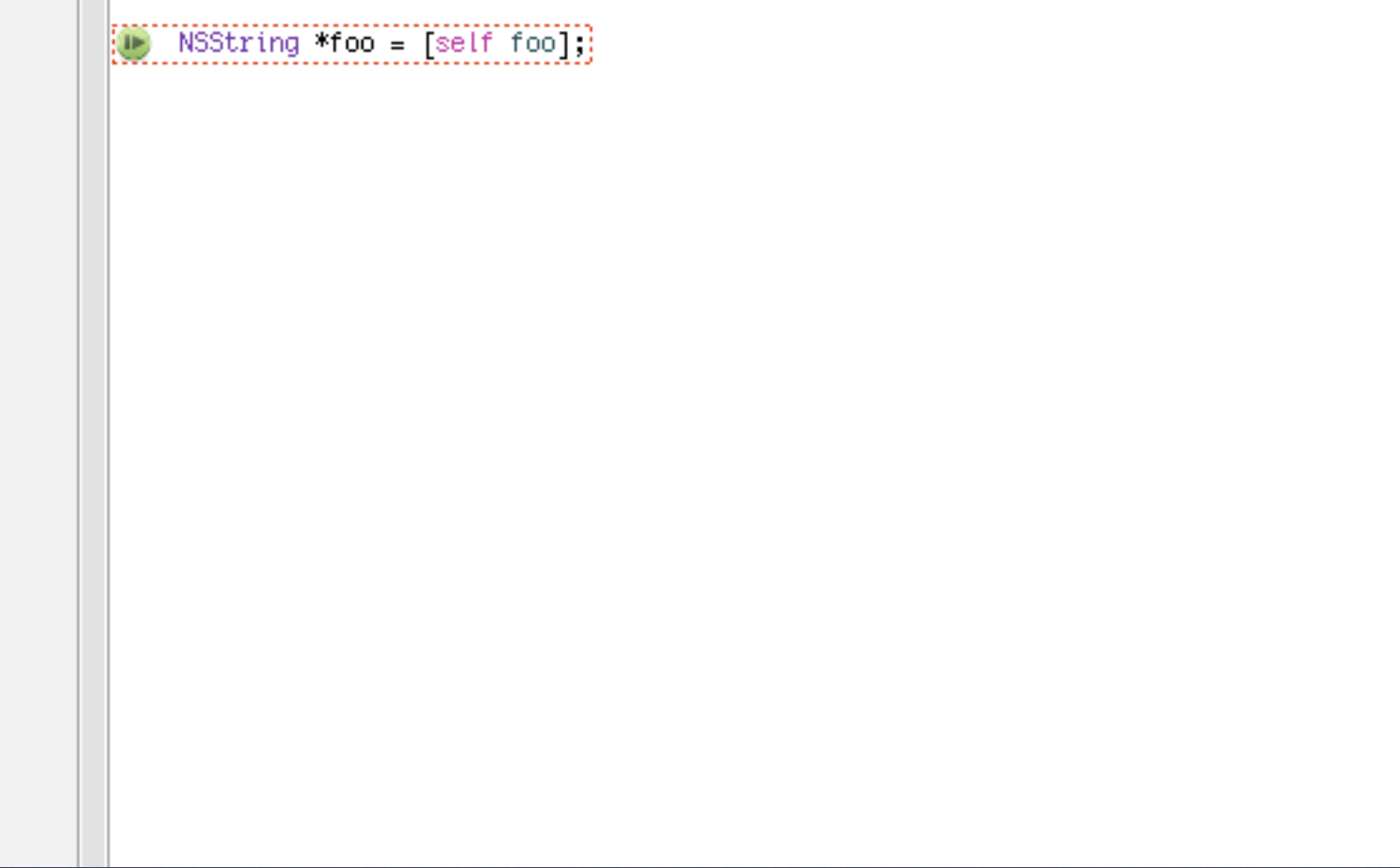
```
NSString *foo [self foo];
```

Evaluate Expression

```
NSString *foo = [self foo];
```

▶	NSString *	[self foo]	0x2030	foo
---	------------	------------	--------	-----

Evaluate Expression



```
▶ NSString *foo = [self foo];
```

“Continue To Here” Step Control

```
NSString *foo [self foo];
```

“Step Into” Step Control

# FAQ

- Build for Debug!
- STABS > DWARF!
- Disable lazy symbol loading?
- Disable data formatters?
- Create and / or customize executables!

# Summary

- Run vs. Debug? Run + Debug!
- Breakpoints can often replace logs
- Breakpoint management
- Executable management
- Break old habits

# References

- Xcode User Guide > Debugging Programs
- Technical Note TN2124: Debugging Magic
- “man gdb”

Q & A