



Debugging.TV

Frame 0x0B

Presenter: Dmitry Vostokov

MEMORY DUMP ANALYSIS SERVICES

DumpAnalysis.com

Including Crash and Hang Analysis Audit, Training and Seminars

Sponsors

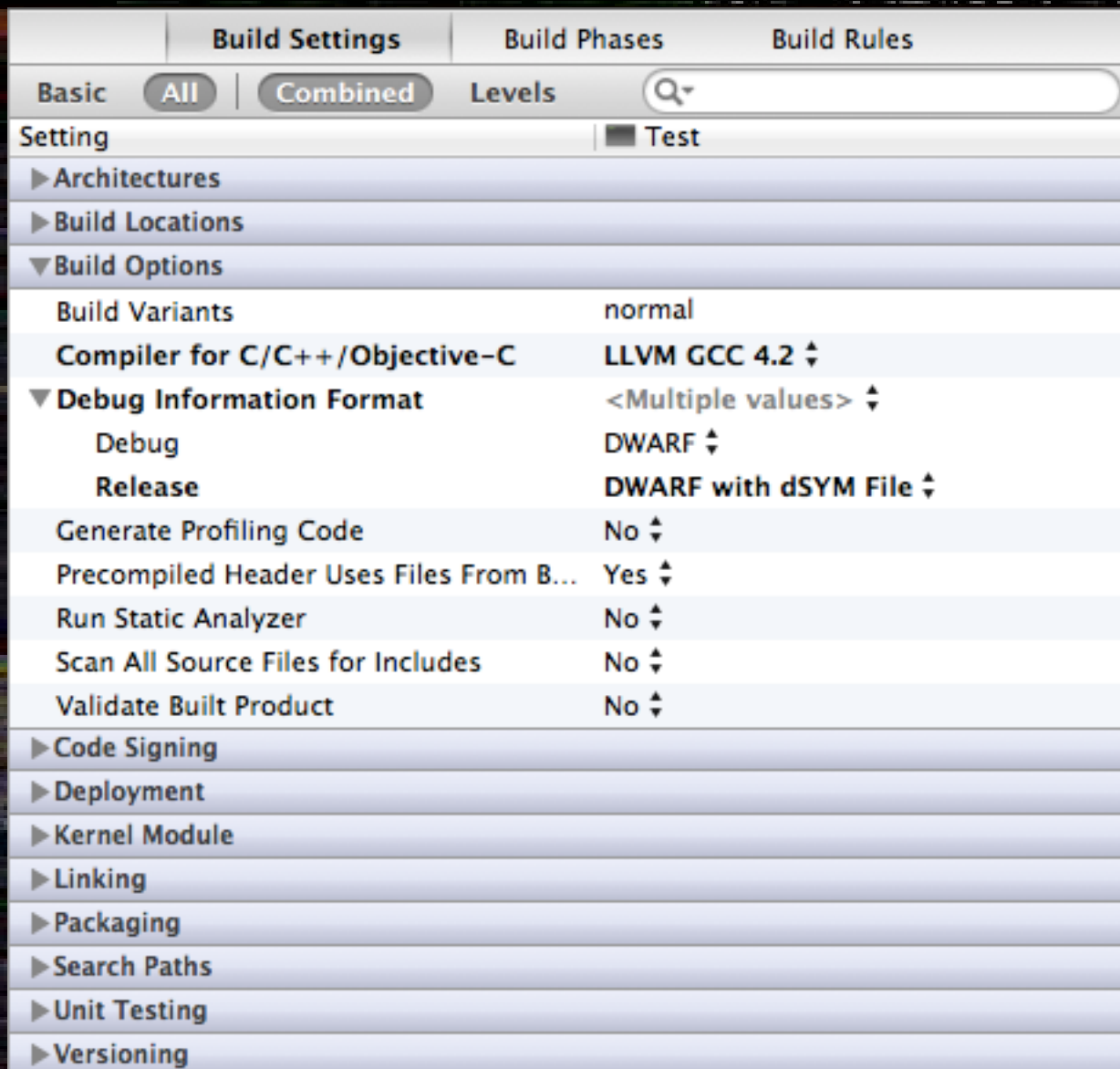
OPENTASK

Iterative and Incremental Publishing

Topics

- Xcode build options
- External symbol files
- GDB: Stack trace w/o symbols
- GDB: Loading and applying symbols

Xcode Build Options



The screenshot shows the Xcode Build Settings window. At the top, there are tabs for 'Build Settings', 'Build Phases', and 'Build Rules'. Below these, there are sub-tabs for 'Basic', 'All', 'Combined', and 'Levels', along with a search field. The 'Setting' section is expanded to show a list of build options. A yellow arrow points to the 'Release' setting, which is set to 'DWARF with dSYM File'.

Setting	Value
Build Variants	normal
Compiler for C/C++/Objective-C	LLVM GCC 4.2
Debug Information Format	<Multiple values>
Debug	DWARF
Release	DWARF with dSYM File
Generate Profiling Code	No
Precompiled Header Uses Files From B...	Yes
Run Static Analyzer	No
Scan All Source Files for Includes	No
Validate Built Product	No

Xcode Deployment Options

Alternate Permissions Files	
Combine High Resolution Artwork	No ↕
Deployment Location	No ↕
▶ Deployment Postprocessing	Yes ↕
Install Group	staff
Install Owner	DumpAnalysis
Install Permissions	u+w,go-w,a+rX
Installation Build Products Location	/tmp/Test.dst
Installation Directory	/usr/local/bin
Mac OS X Deployment Target	Mac OS X 10.7 ↕
Pre-install Requirements Property List	
Skip Install	No ↕
▼ Strip Debug Symbols During Copy	<Multiple values> ↕
Debug	No ↕
Release	Yes ↕
▼ Strip Linked Product	<Multiple values> ↕
Debug	No ↕
Release	Yes ↕
Strip Style	All Symbols ↕
Use Separate Strip	No ↕
▶ Kernel Module	



Xcode Code Generation Options

Setting	Test
▼ LLVM GCC 4.2 - Code Generation	
Accelerated Objective-C Dispatch	Yes ↕
Call C++ Default Ctors/Dtors in Objec...	Yes ↕
Enable SSE3 Extensions	No ↕
Enable SSE4.1 Extensions	No ↕
Enable SSE4.2 Extensions	No ↕
Enable Supplemental SSE3 Instructions	No ↕
Enforce Strict Aliasing	No ↕
Feedback-Directed Optimization	Off ↕
► Generate Debug Symbols	Yes ↕
Generate Position-Dependent Code	No ↕
Generate Test Coverage Files	No ↕
Inline Methods Hidden	Yes ↕
Instrument Program Flow	No ↕
Kernel Development Mode	No ↕
Level of Debug Symbols	Default [default, -gstabs+ -felimina... ↕
Link-Time Optimization	No ↕
Make Strings Read-Only	Yes ↕
No Common Blocks	No ↕
Objective-C Garbage Collection	Unsupported ↕
Optimization Level	None [-O0] ↕



GDB Output

```
(gdb) bt
```

```
#0 0x000000010d3b0e90 in ?? ()  
#1 0x000000010d3b0ea9 in ?? ()  
#2 0x000000010d3b0ec4 in ?? ()  
#3 0x000000010d3b0e74 in ?? ()
```

```
(gdb) maintenance info sections
```

```
[...]
```

```
Core file:
```

```
  `~/cores/core.262', file type mach-o-le.
```

```
  0x000000010d3b000->0x000000010d3b1000 at 0x00001000: LC_SEGMENT. ALLOC LOAD
```

```
CODE HAS_CONTENTS
```

```
[...]
```

```
(gdb) add-symbol-file ~/Documents/Work/Test.sym 0x000000010d3b0000
```

```
add symbol table from file "/Users/DumpAnalysis/Documents/Work/Test.sym" at
```

```
  LC_SEGMENT.__TEXT = 0x10d3b0000
```

```
(y or n) y
```

```
Reading symbols from /Users/DumpAnalysis/Documents/Work/Test.sym...done.
```

```
(gdb) bt
```

```
#0 0x000000010d3b0e90 in bar () at main.c:15  
#1 0x000000010d3b0ea9 in foo () at main.c:20  
#2 0x000000010d3b0ec4 in main (argc=1,  
  argv=0x7fff6cfafbf8) at main.c:25
```

!Ad Hardcore Technical Support Training

April 2, 2012: [Introduction to Software Narratology](#) (FREE)

April 11-16, 2012: [Accelerated Windows Memory Dump Analysis](#)

April 20-23, 2012: [Advanced Windows Memory Dump Analysis](#)

April 27-30, 2012: [Accelerated Software Trace Analysis](#)

[Accelerated Mac OS X Core Dump Analysis](#)

Forthcoming: [Linux Core Dump Analysis](#)

[Deep Down C++](#)

[Training Schedule](#)

Debugging.TV