

FILECHECK

*FOLLIES*

**- or -**

*Did you test your test?*

Paul Robinson

Sony Interactive Entertainment

***RIPPED FROM THE HEADLINES  
COMMIT LISTS!***



# Order in the Court!



- Make sure 'bar' not preceded by 'foo'

```
> cat Order-1.txt
// CHECK-NOT: foo {{.*}} bar
// CHECK:      bar
> echo 'foo bar' | FileCheck Order-1.txt
> # Oops it passed.
> # The CHECK-NOT is done after the CHECK, and
> # the CHECK delimits the range examined by CHECK-NOT.
```

# Order in the Court!



- Want 'foo' and 'bar' on different lines

```
> cat Order-2.txt
```

```
// CHECK:          foo
```

```
// CHECK-SAME-NOT: bar
```

```
// CHECK:          bar
```

```
> echo 'foo bar' | FileCheck Order-2.txt
```

```
Order-2.txt:2:9: error: unsupported -NOT combo on prefix  
'CHECK'
```

```
// CHECK-SAME-NOT: bar
```

```
^
```

```
> # Can't combine suffixes!
```

# A (Pre)Fixed Race



- Case X sees 'foo' and case Y sees 'foo1'

```
> cat Prefix-1.txt
```

```
// X: foo
```

```
// Y: foo1
```

```
> # Give Y's output to X's checks, probably should fail:
```

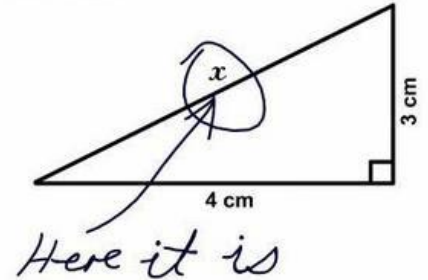
```
> echo 'foo1' | FileCheck Prefix-1.txt -check-prefix=X
```

```
> # Oops that passed.
```

```
> # So: Need to make the CHECK patterns distinct.
```

# A LIT-eral Check

3. Find x.



Ocular Trauma - by Wade Clarke ©2005

- Keywords? What keywords?

```
> cat LIT-0.txt
```

```
I'm sad the Giants couldn't win the World Series.
```

```
> cat LIT-1.txt
```

```
// UNSUPPORTED: win
```

```
> cat LIT-0.txt |
```

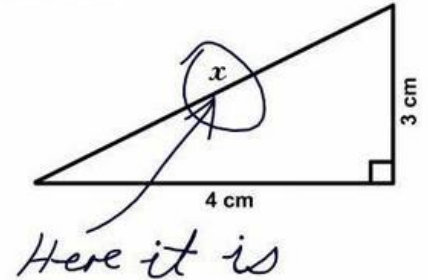
```
FileCheck LIT-1.txt -check-prefix=UNSUPPORTED
```

```
> # Wait--did that work?
```

```
> # FileCheck won't object to Lit keyword as prefix.
```

# A LIT-eral Check

3. Find x.



Ocular Trauma - by Wade Clarke ©2005

- Keywords? What keywords?

```
> cat LIT-2.txt
// RUN: cat LIT-0.txt |
    FileCheck LIT-2.txt -check-prefix=UNSUPPORTED
// UNSUPPORTED: win
Somebody running on Windows...
> llvm-lit.py LIT-2.txt
...
UNSUPPORTED: LIT-2.txt
Testing Time: 0.03s
  Unsupported Tests   : 1
> # Oops--doesn't test correctly under Lit
> # So: Don't use Lit keywords as prefixes.
```

# LABEL Carefully



```
> cat Label-0.txt
; Test for the label example.
t:
    part 1
x:
    part 2
```

```
> cat Label-1.txt
// CHECK-LABEL: t
// CHECK:          part 1
// CHECK-LABEL: x
// CHECK:          part 2
```



# LABEL Carefully



```
> cat Label-0.txt | FileCheck Label-1.txt
Label-1.txt:2:11: error: expected string not found in
input
// CHECK: part 1
      ^
<stdin>:1:4: note: scanning from here
; Test for the label example.
      ^
> # The 't' in Test isn't a label... but it matched,
> # and the second -LABEL matched 'x' in 'example.'
> # (CHECK-LABEL executes before CHECK.)
> # So: Include the punctuation on -LABEL.
```

# -SAME is Different



- Sometimes a check line gets too long...

```
> cat Same-1.txt
// OLD:          -o {{.*}}.o {{.*}} -x c++
// NEW:          -o {{.*}}.o
// NEW-SAME:    -x c++
> echo '-o foo.o -O -x c++ llvm.org' |
  FileCheck Same-1.txt -check-prefix=OLD
> # Original check does work.
```

# -SAME is Different



- Sometimes a check line gets too long...

```
> cat Same-1.txt
// OLD:      -o {{.*}}.o {{.*}} -x c++
// NEW:      -o {{.*}}.o
// NEW-SAME: -x c++
> echo '-o foo.o -O -x c++ llvm.org' |
  FileCheck Same-1.txt -check-prefix=NEW
Same-1.txt:3:14: error: expected string not found in input
// NEW-SAME: -x c++
                ^

<stdin>:1:26: note: scanning from here
-o foo.o -O -x c++ llvm.org
                ^

> # Wildcard matches more than intended!
```

# What's In a Name?



- ‘foo’ and ‘bar’ are obviously not real names
  - Unless your last name is Foote or Lebar
- Try it: top-level directory named ‘foobar’
  - Checkout llvm, clang, compiler-rt
  - ‘ninja check-all’ shows 6 failures
- Not actually a FileCheck problem but seemed worth mentioning

# Are You Follie Fodder?

- CHECK-NOT is not the first check
- Be wary of matching prefixes
- Avoid LIT keywords
- CHECK-LABEL doesn't require labels
- CHECK-SAME is different (with wildcards)
- 'foo' and 'bar' are (part of) real names

# Order in the Court!



- Want 'foo' and 'bar' on different lines

```
> cat Order-3.txt
// CHECK:      foo
// CHECK-NOT:  bar
// CHECK:      {{$}}
// CHECK:      bar
> echo 'foo bar' | FileCheck Order-3.txt
<stdin>:1:6: error: CHECK-NOT: string occurred!
'foo bar'
  ^
Order-3.txt:2:15: note: CHECK-NOT: pattern specified here
// CHECK-NOT:  bar
                ^
> # YES!!!
```

# A (Pre)Fixed Race



- Case X sees 'foo' and case Y sees 'foo1'

```
> cat Prefix-1.txt
```

```
// X: foo
```

```
// Y: foo1
```

```
> echo 'foo1' | FileCheck Prefix-1.txt -check-prefix=X
```

```
> # Oops that passed.
```

```
> cat Prefix-2.txt
```

```
// X: foo{{ }}
```

```
// Y: foo1
```

```
> # This will work.
```