

-Wall found compilation errors

Aditya Kumar
Facebook

Enable compiler warnings with -Wall -Werror

- Compiler warnings are precise static analysis tools
- -Wall enables most relevant compiler warnings
 - Widely used warnings, well tested, less chance of false positives
- Adding -Werror fails the compilation

-Wpointer-bool-conversion

```
bool foo();

void bar(bool x)
{
    if (foo && x) {
        // ...
    }
}
```

error: address of function 'foo' will always evaluate to 'true'

[-Werror,-Wpointer-bool-conversion]

-Wself-assign

```
class foo {  
public:  
    foo(int a) {  
        _a = _a;  
    }  
}
```

```
private:  
    int _a;  
};
```

error: assigning field to itself [-Werror,-Wself-assign]

PS: Using IDE can help because of syntax highlighting.

-Winfinite-recursion

```
// library1.h:
```

```
#define _aligned_free free
```

```
// folly/Memory.h
```

```
void _aligned_free(void *p) {  
    free(p);  
}
```

```
// test.cpp:
```

```
#include<library1.h>
```

```
#include<folly/Memory.h>
```

```
int main()
```

```
{
```

```
    int *p2 = _aligned_alloc(64, 10*sizeof *p2);
```

```
    // ...
```

```
    _aligned_free(p2);
```

```
}
```

error: all paths through this function will call itself [-Werror,-Winfinite-recursion]

-Wunused-labels

```
#include<string>
using namespace std;
void foo() {
    std:
        string a("asdfasdf");
}
```

Typing too fast?

```
void checkSeverity(SeverityType T) {
    switch(T) {
        Normal: return; // Do nothing
        High: warn(); return;
        Critical: cleanup(); return;
        default: return;
    }
}
```

Case against case statement?

error: unused label 'std' [-Werror,-Wunused-label]

-Woverloaded-virtual

```
// Base.h
class Base
{
public:
    virtual void foo(unsigned a);
};
```

```
#include<Base.h>
// Derived.h
class Derived : Base
{
public:
    void foo(int a);
};
```

error: 'Derived::foo' hides overloaded virtual function [-Werror,-Woverloaded-virtual]
note: hidden overloaded virtual function 'Base::foo' declared here: type mismatch at
1st parameter ('unsigned int' vs 'int')

How to enable -Wall for a large codebase

- Directory level enabling is tedious
- Build applications with -Wall -Werror and continue the build to collect as many errors (e.g., -k flag in make)
- With some bash magic get the list of all the warnings
 - e.g., `grep 'error:' build.log | sed 's/.*Werror,//g' | sed 's/\]//g' | sort -u`
- Disable those warnings (by adding their -Wno versions)
- Keep fixing one warning at a time...
- Disable warnings in the build target where changing the code isn't possible
 - Use pragma (e.g., `#pragma clang diagnostic ignored "-Woverloaded-virtual"`) if necessary.
- Some warnings appear way more than others
 - -Wreorder
 - -Wpessimizing-move
 - -Woverloaded-virtual