

The results are in! [See the 2018 Developer Survey results.](#) »

×

Generate calling graph for C++ code

I'm trying to generate calling graph with which to find out all the possible execution paths that are hitting a particular function (so that I don't have to figure out all the paths manually, as there are many paths that lead to this function). For instance:

```
path 1: A -> B -> C -> D
path 2: A -> B -> X -> Y -> D
path 3: A -> G -> M -> N -> O -> P -> S -> D
...
path n: ...
```

I have tried Codeviz and Doxygen, somehow both results show nothing but callees of target function, D. In my case, D is a member function of a class whose object will be wrapped within a smart pointer. Clients will always obtain the smart pointer object through a factory in order to invoke D.

Does anyone know how to achieve this?

c++

call-graph

edited Nov 14 '15 at 13:46

 m.s.

11.9k 5 33 57

asked Mar 21 '11 at 4:09

 shiouming

736 1 9 21

6 Answers

```
static void D() { }
static void Y() { D(); }
static void X() { Y(); }
static void C() { D(); X(); }
static void B() { C(); }
static void S() { D(); }
static void P() { S(); }
static void O() { P(); }
static void N() { O(); }
static void M() { N(); }
static void G() { M(); }
static void A() { B(); G(); }

int main() {
    A();
}
```

Then

```
$ clang++ -S -emit-llvm main1.cpp -o - | opt -analyze -dot-callgraph
$ dot -Tpng -ocallgraph.png callgraph.dot
```

Yields some shiny picture (there is an "external node", because `main` has external linkage and might be called from outside that translation unit too):

