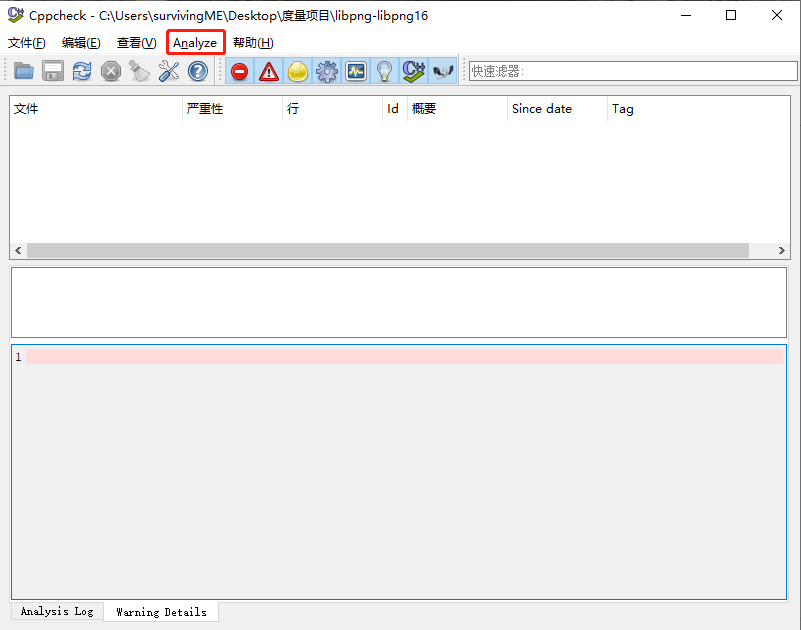
Cppcheck

* libpng

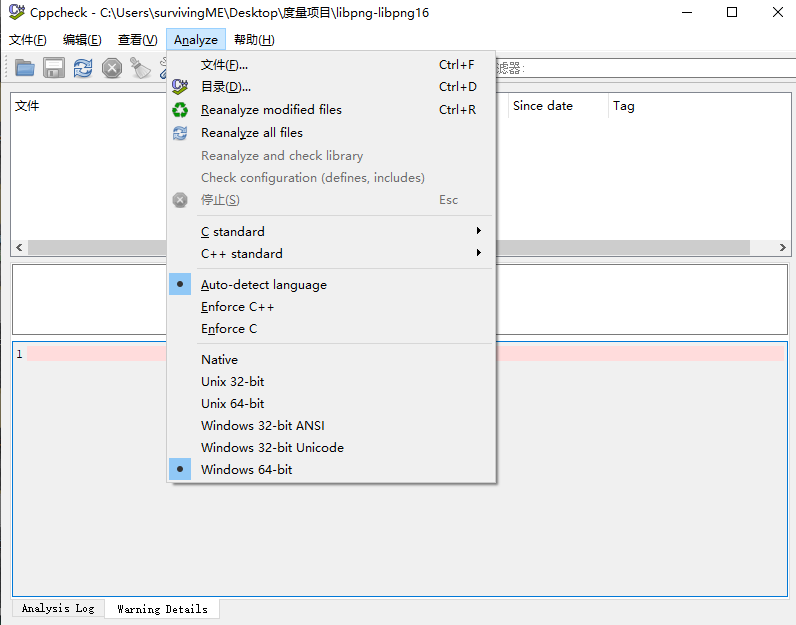
1. 克隆项目到本地：

git clone https://github.com/glennrp/libpng.git

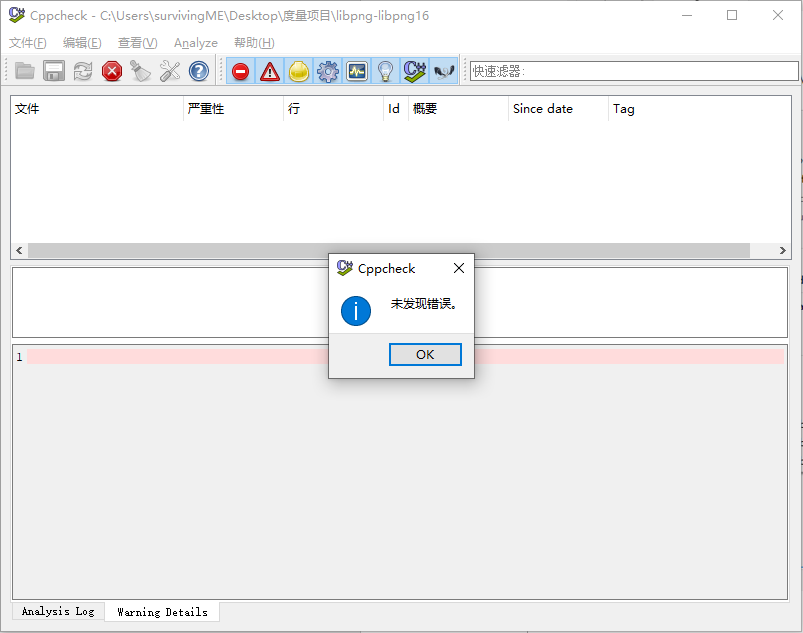
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

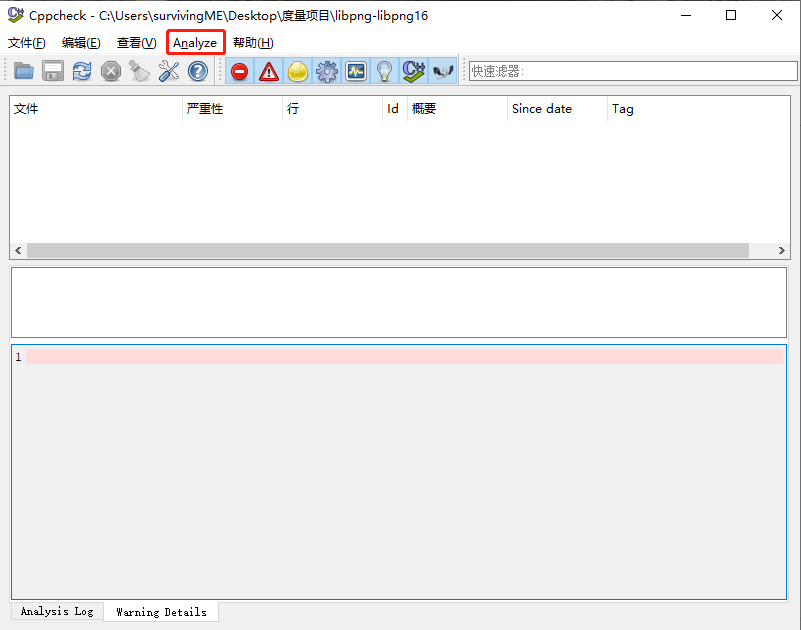


* libjpeg-turbo

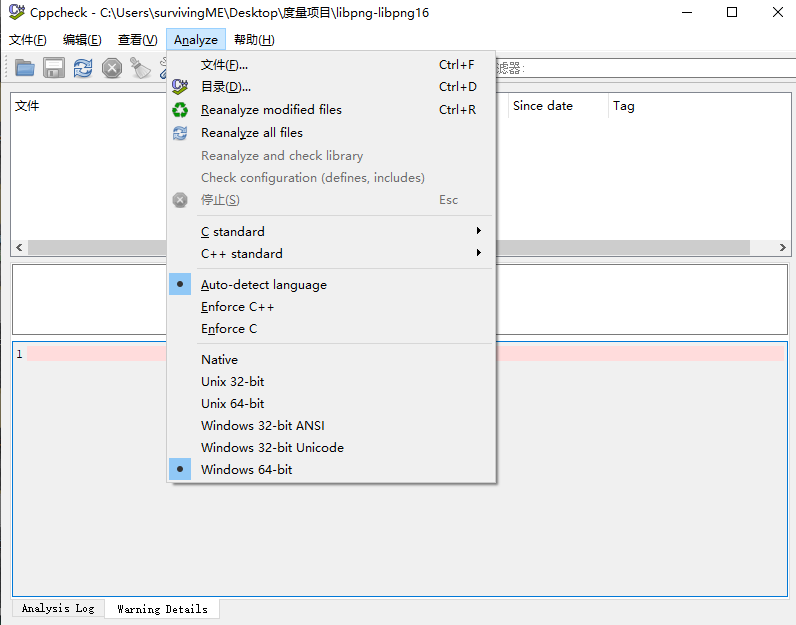
1. 克隆项目到本地

git clone https://github.com/libjpeg-turbo/libjpeg-turbo.git

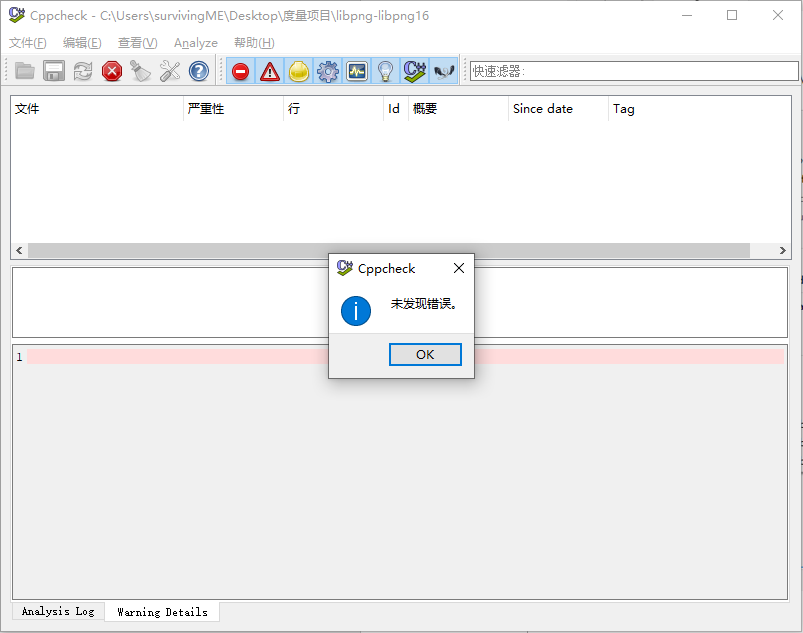
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

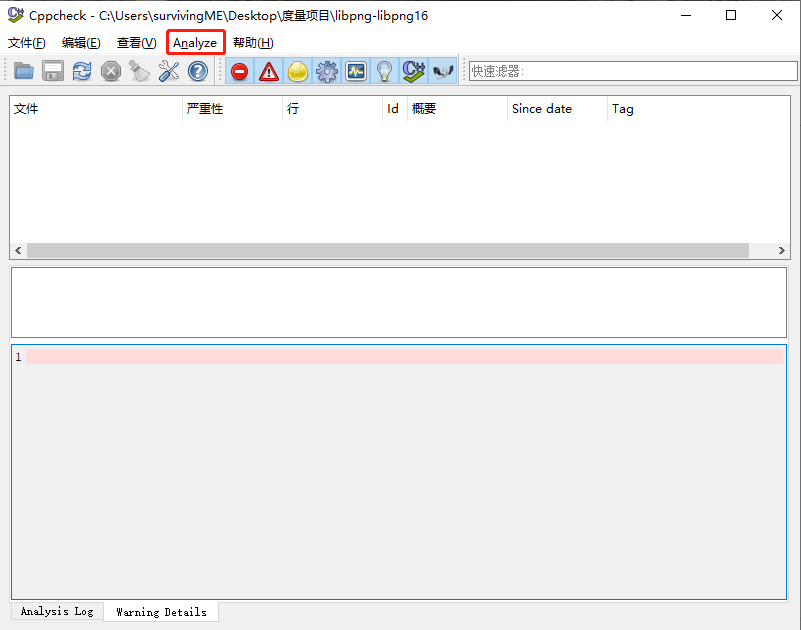


* File programm

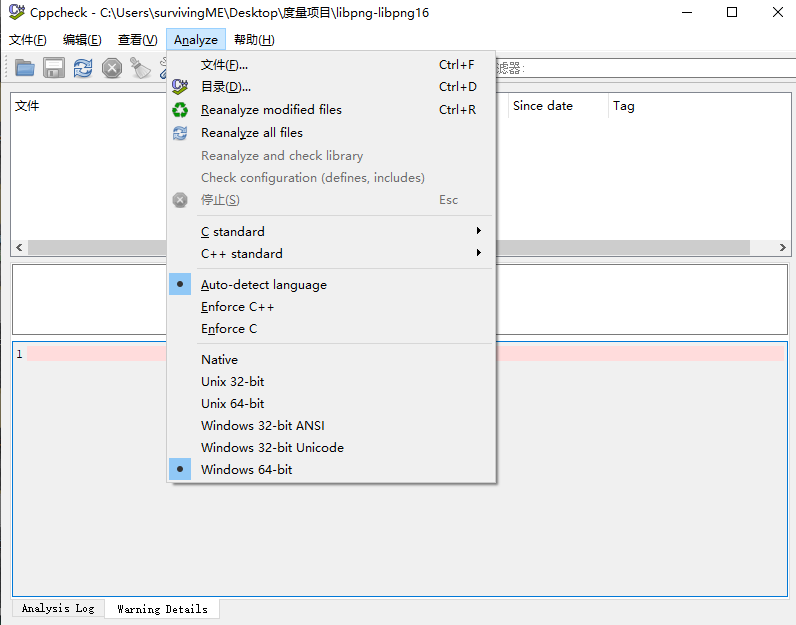
1. 克隆项目到本地

git clone git://git.libssh.org/projects/libssh.git

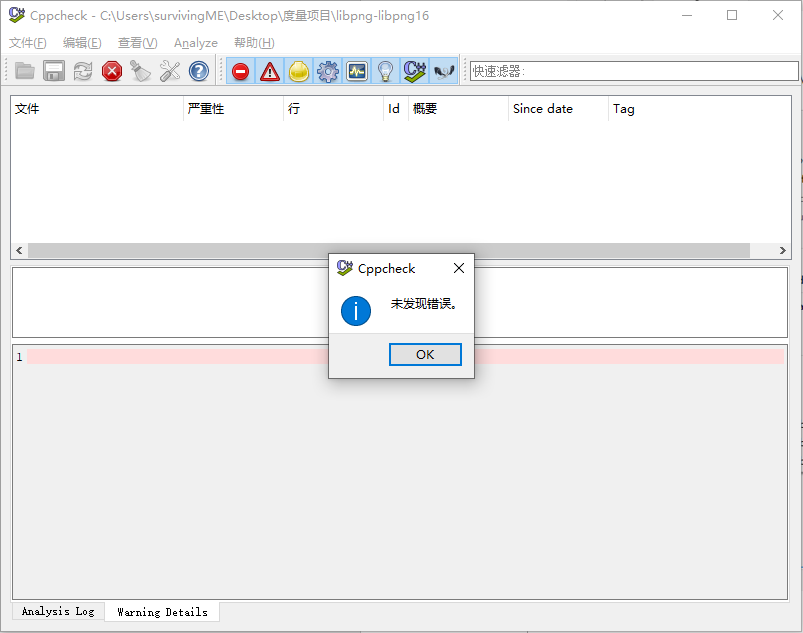
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

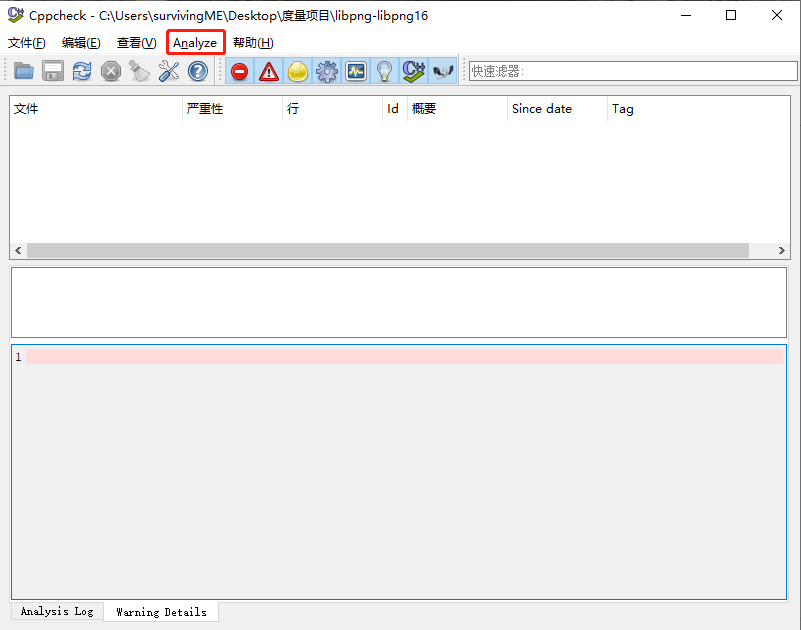


* file programm

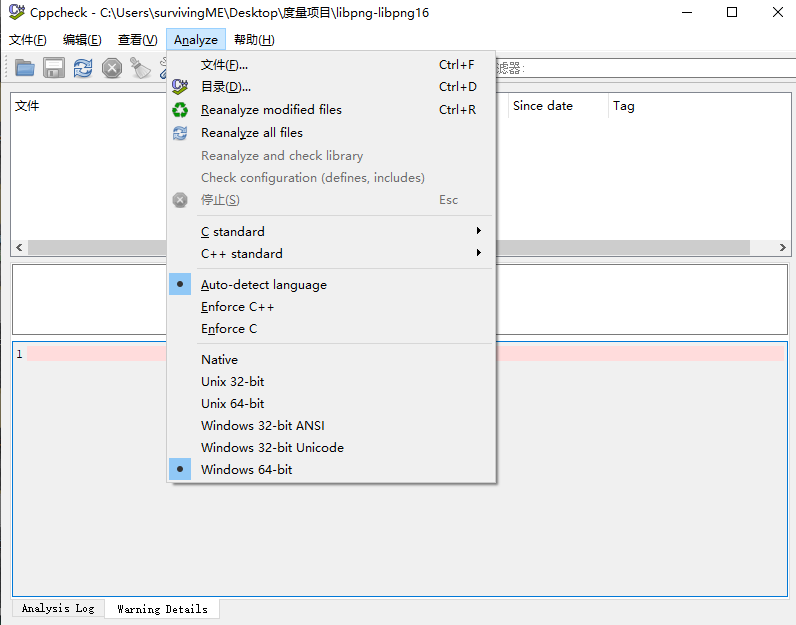
1. 克隆项目到本地：

git clone https://github.com/file/file.git

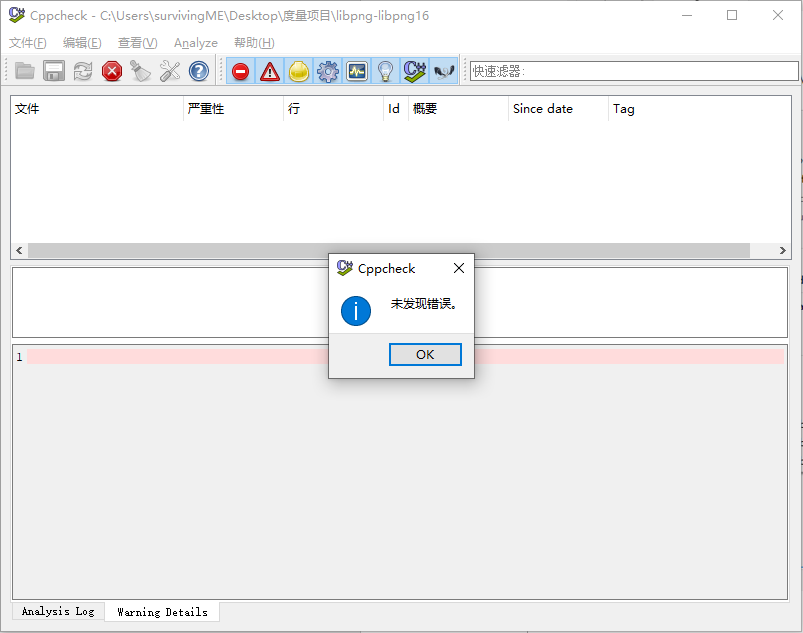
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

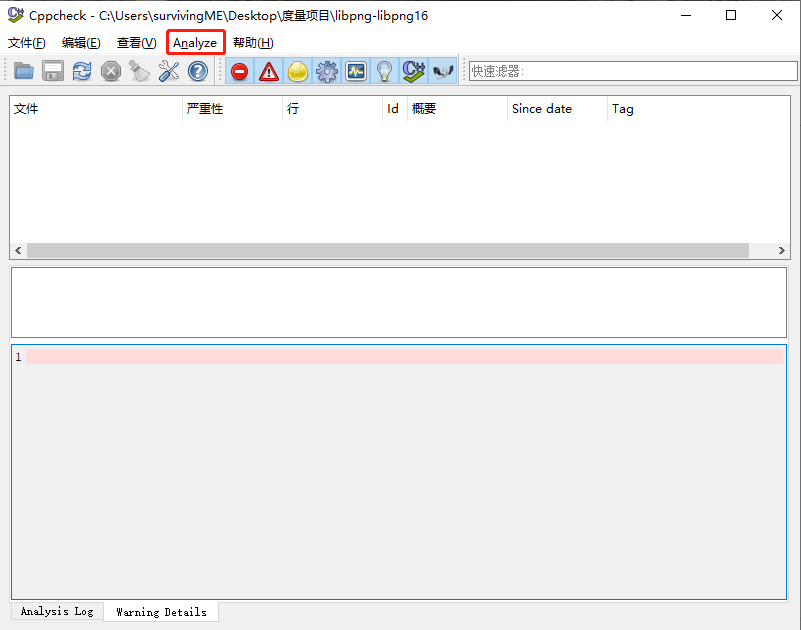


* libconfig

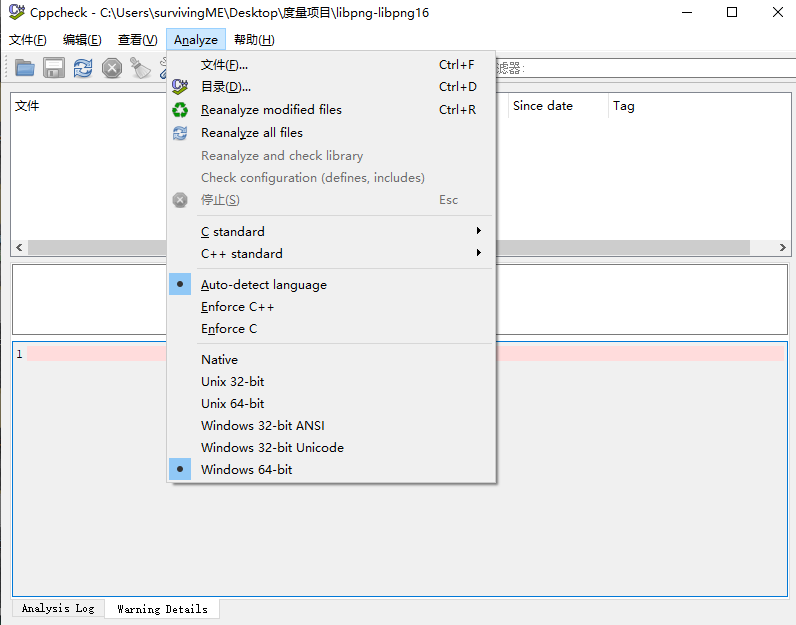
1. 克隆项目到本地：

git clone https://github.com/hyperrealm/libconfig.git

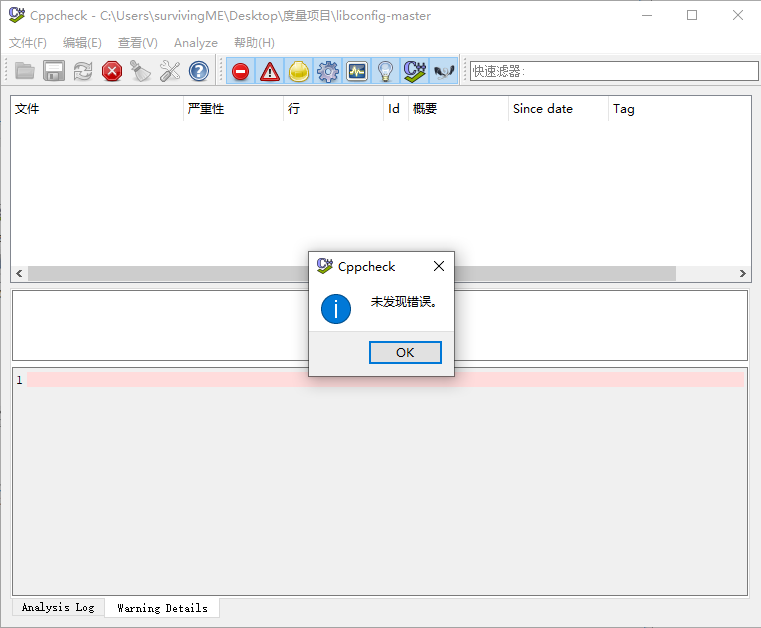
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

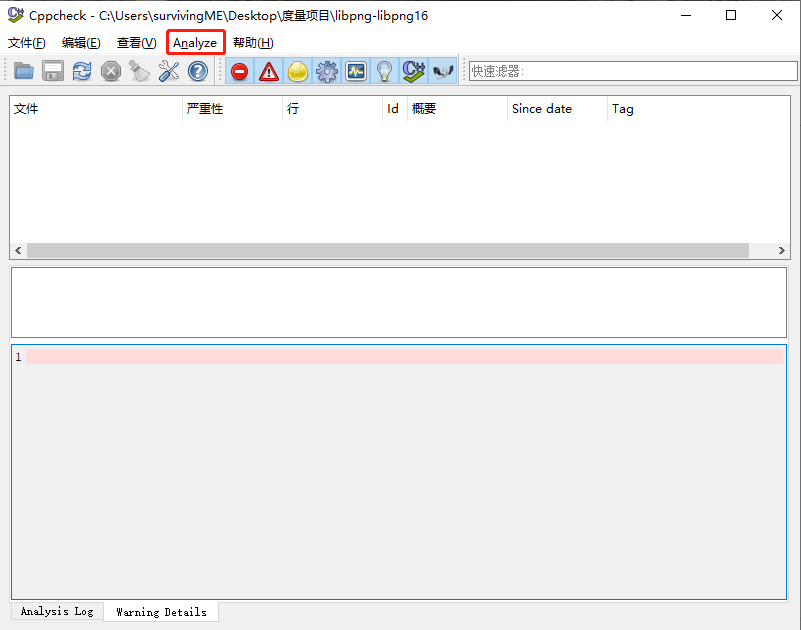


* libcypher-parser

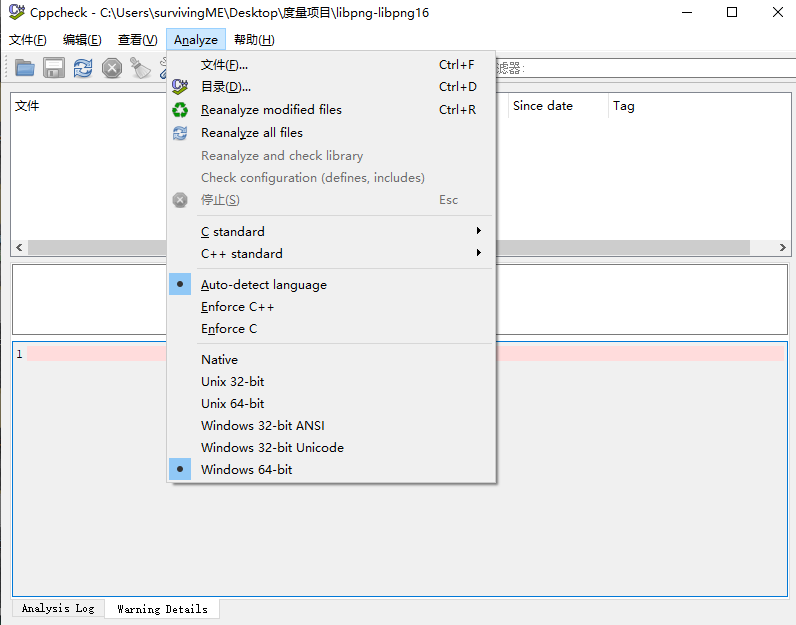
1. 克隆项目到本地：

git clone https://github.com/cleishm/libcypher-parser

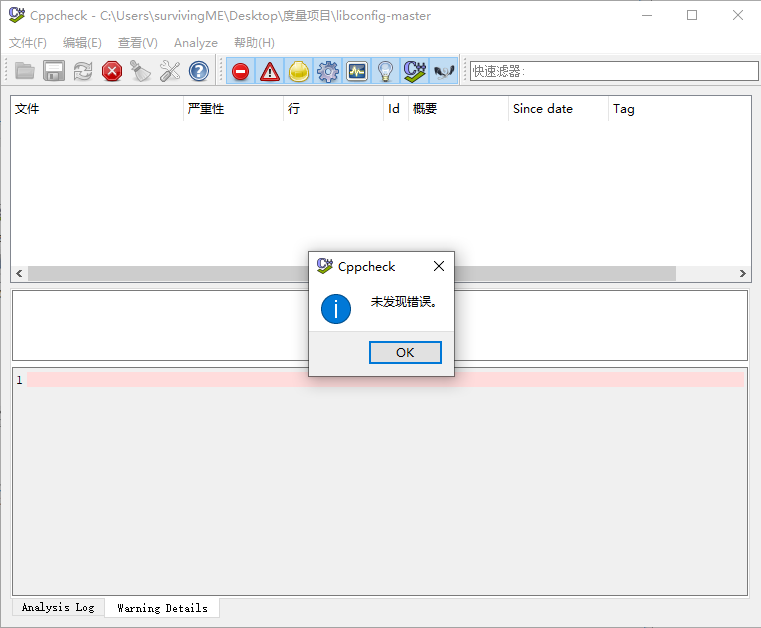
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

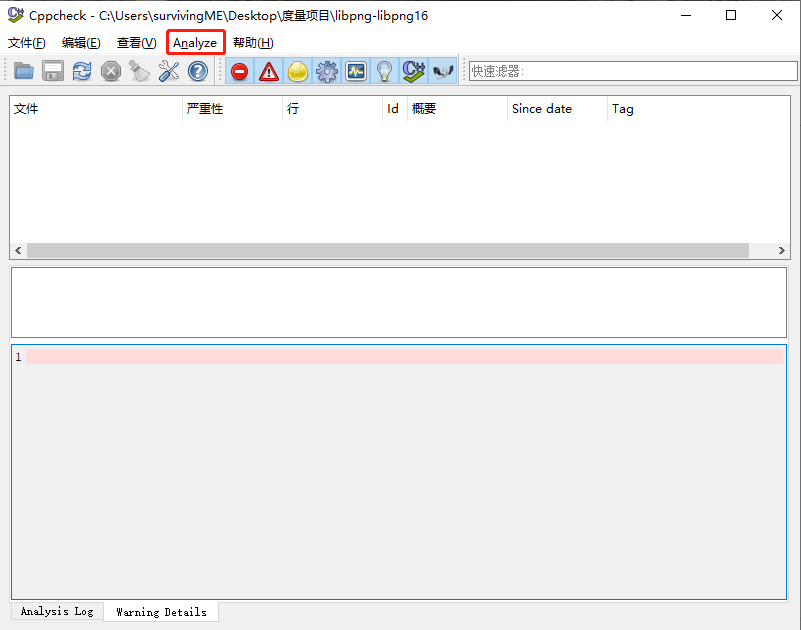


* libbcrypt

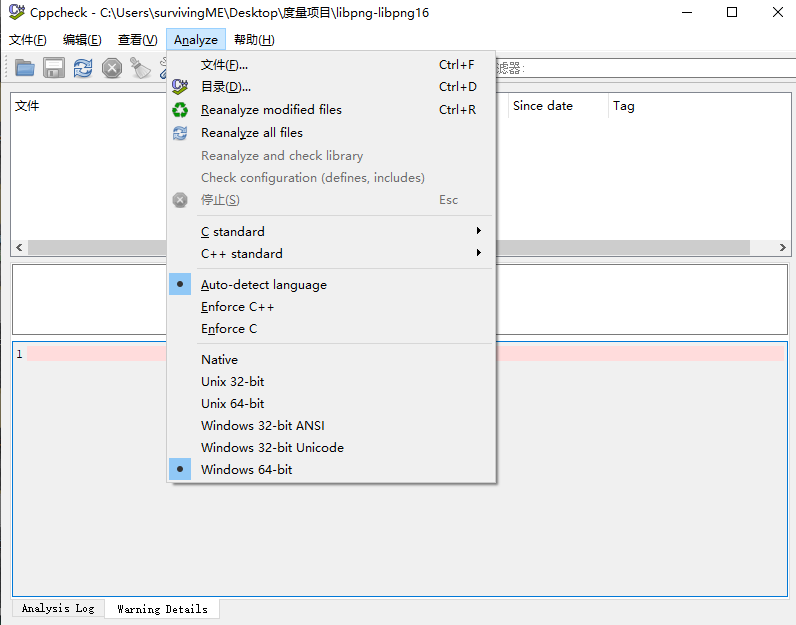
1. 克隆项目到本地：

git clone https://github.com/rg3/libbcrypt.git

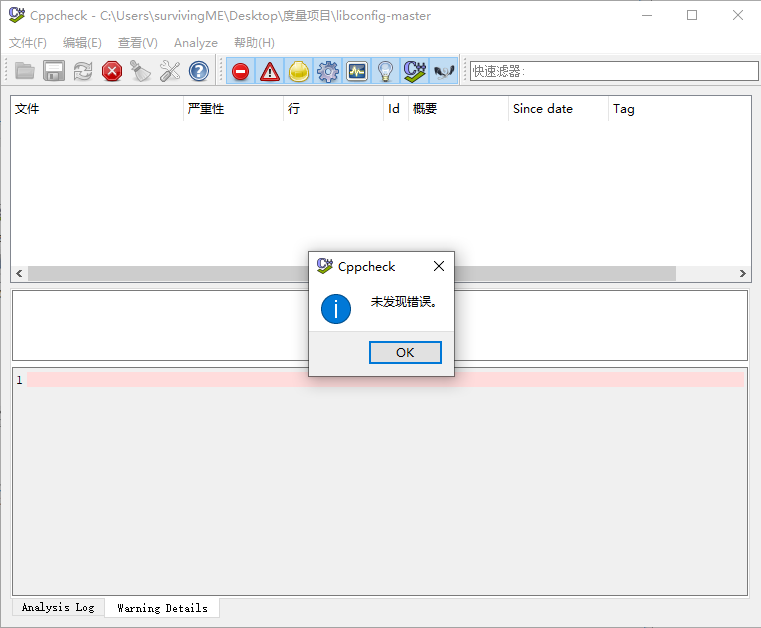
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

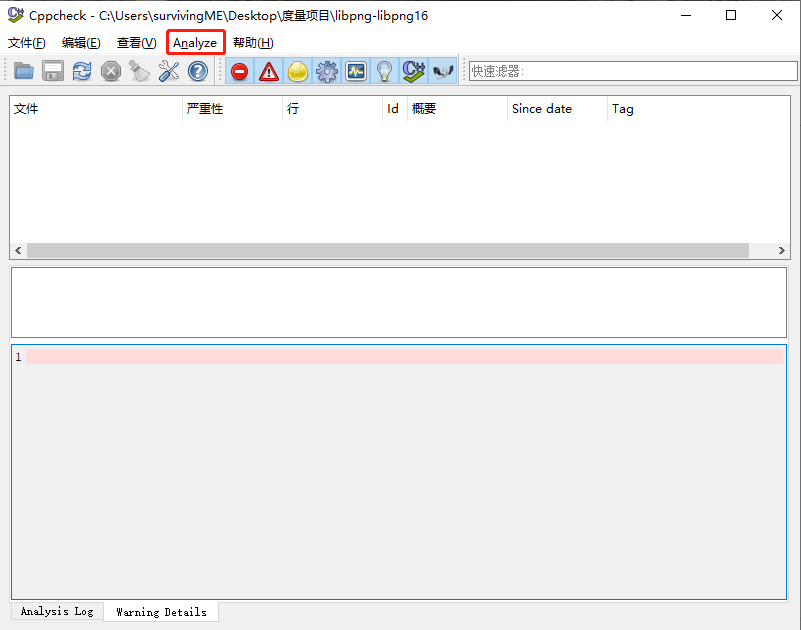


* c-algorithms

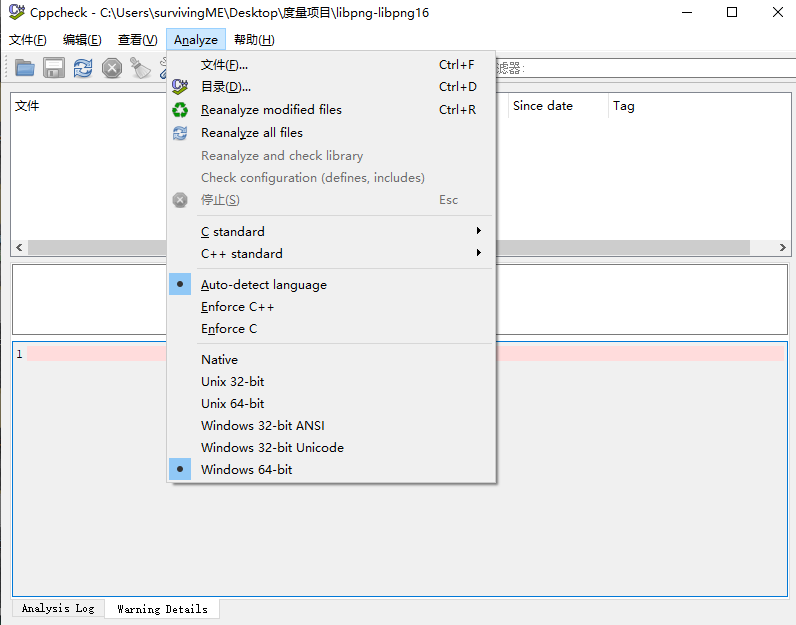
1. 克隆项目到本地：

git clone https://github.com/fragglet/c-algorithms.git

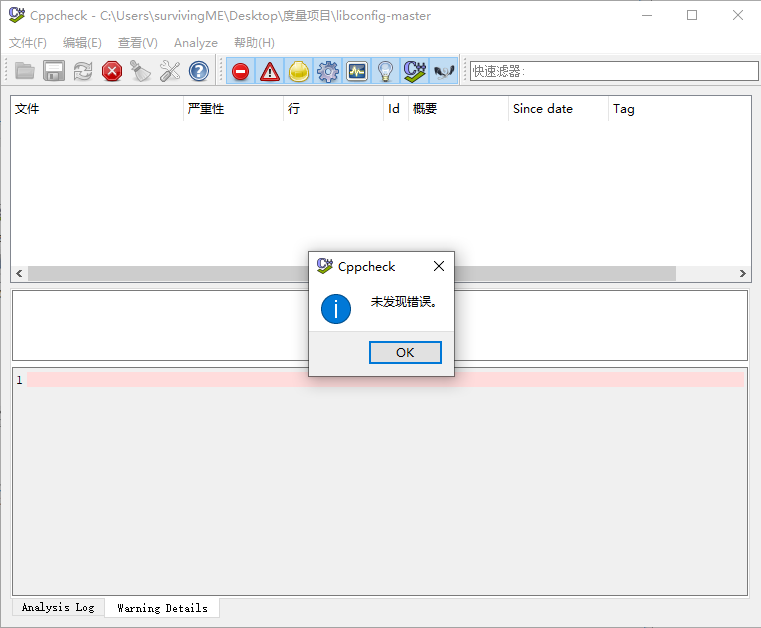
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

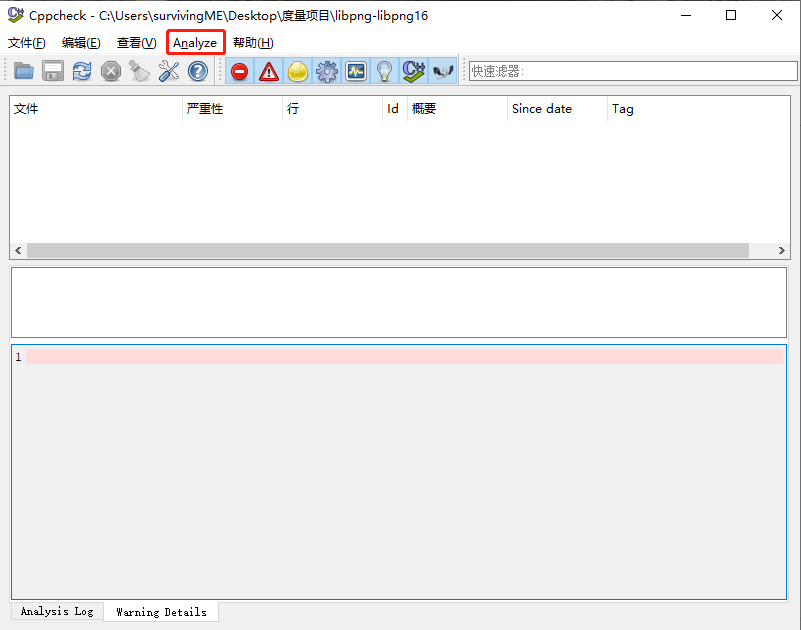


* hashmap

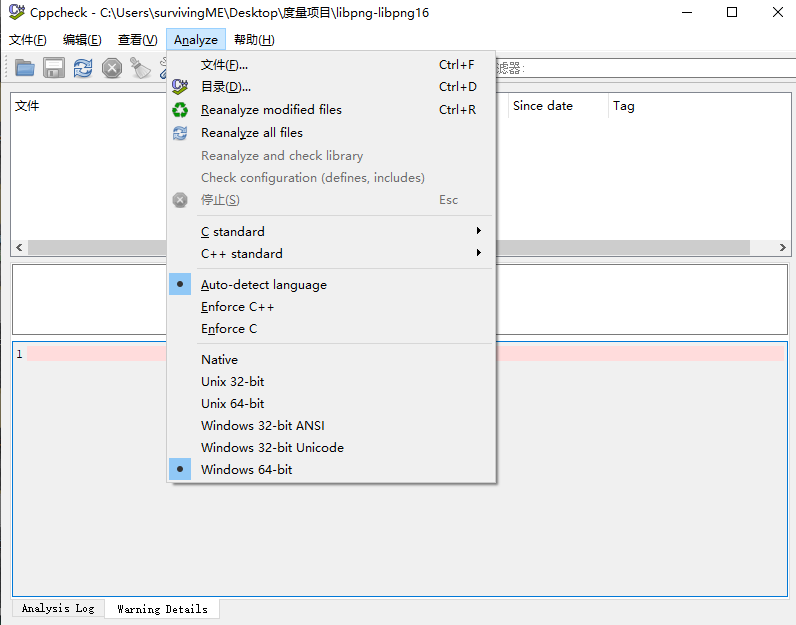
1. 克隆项目到本地：

git clone https://github.com/goossaert/hashmap.git

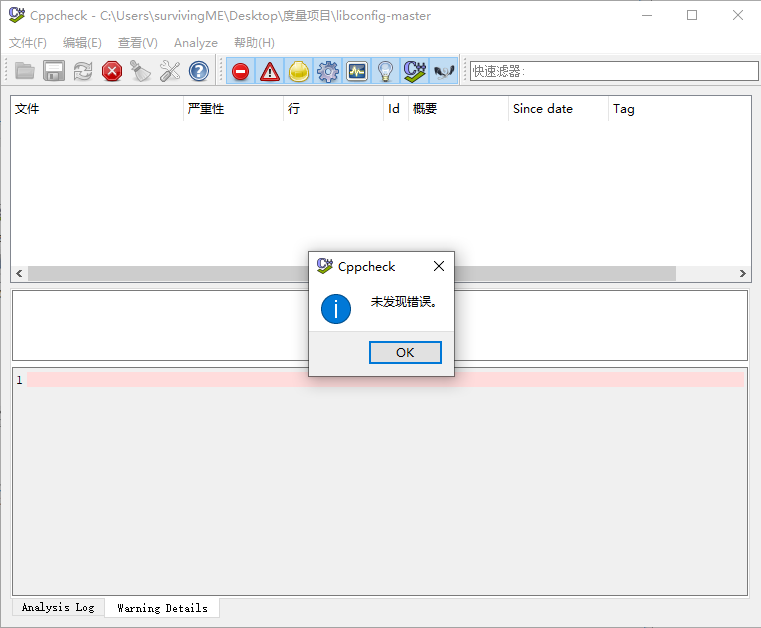
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

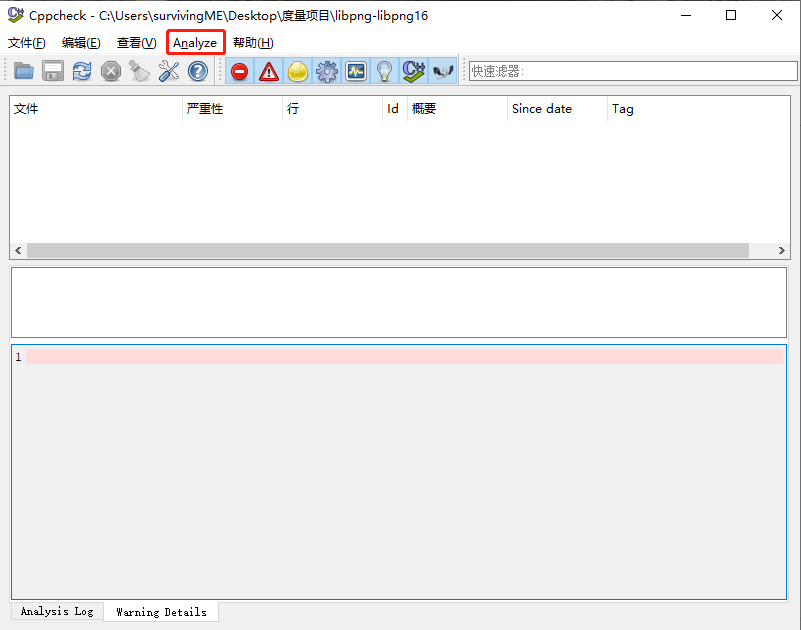


* sds

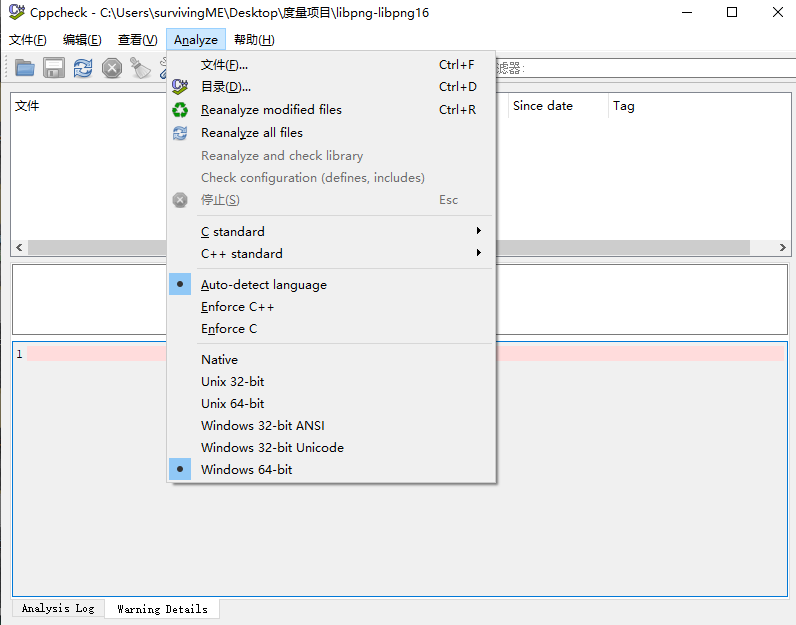
1. 克隆项目到本地：

git clone https://github.com/antirez/sds.git

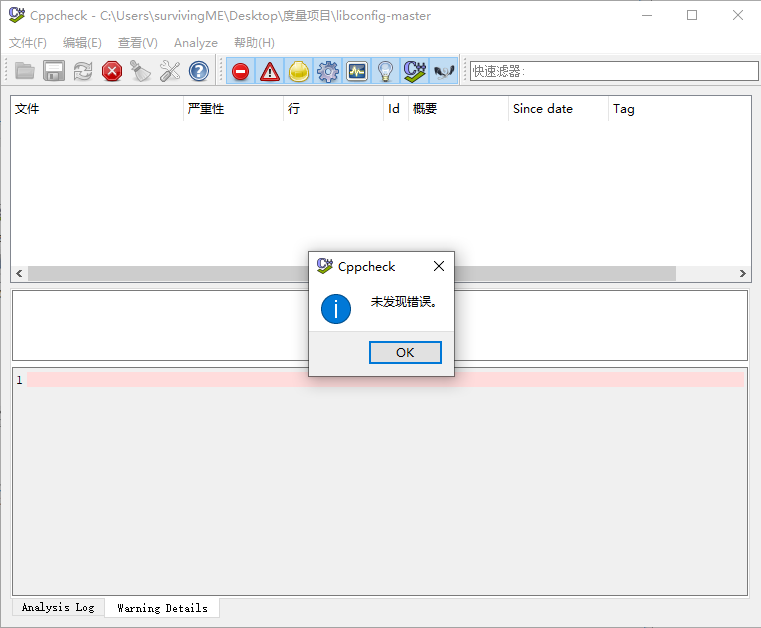
1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

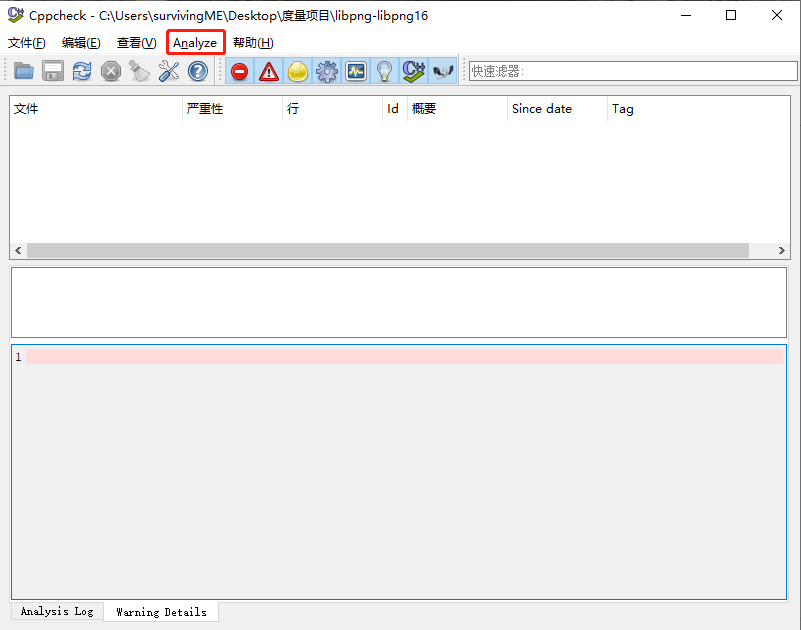


* libexpat

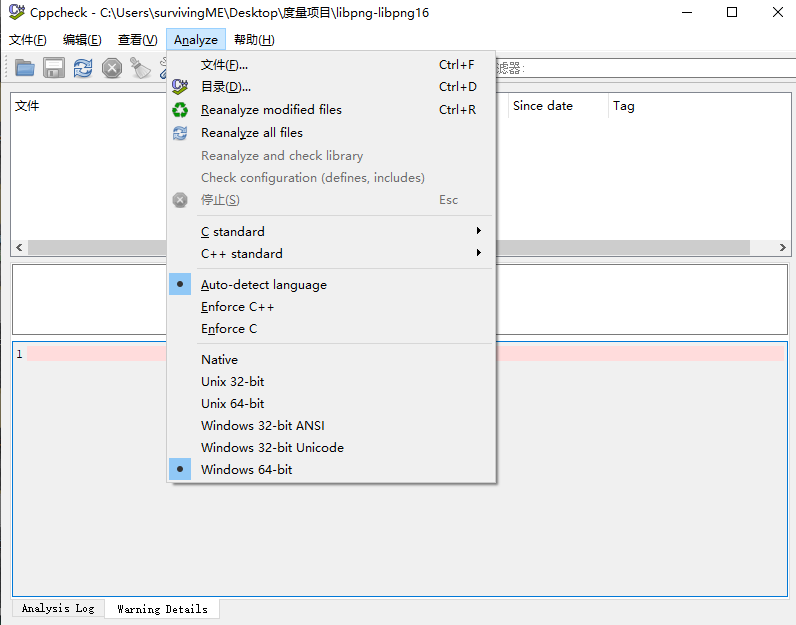
1. 克隆项目到本地：

git clone https://github.com/libexpat/libexpat.git

1. 点击Analyze，准备选择项目进行分析：



1. 选择目录进行检测



1. 检测结果

