Sourcetrail build from source

Build Steps

I was able to build `sourcetrail` from source using Ubuntu 20.04, with the following steps, as mentioned in Sourcetrail how-to-build.

- Download source from github.com/CoatiSoftware/Sourcetrail.
- Install bootstrap (as specified by `Sourcetrail`):
    - Download `bootstrap` from sourceforge.net/projects/boost. See also www.boost.org.
        - `cd ~/Downloads/boost_1_81_0`
        - `./bootstrap.sh --with-libraries=filesystem,program_options,system,date_time`
        - `./b2 --link=static --variant=release --threading=multi --runtime-link=static --cxxflags=-fPIC`
- install Qt5
    - `sudo apt-get install qt5-default qtscript5-dev qttools5-dev-tools libqt5svg5-dev`
- install `ninja`
    - `git clone https://github.com/martine/ninja.git`
    - `cd ninja`
    - `git checkout release`
    - `sudo apt install re2c`   (needed by `configure.py`)
    - `./configure.py --bootstrap`
    - `sudo cp ninja /usr/bin/`
- build `clang/llvm`
        - Download from https://github.com/llvm/llvm-project/releases/   (eg, `llvm-project-15.0.7.src.tar.xz`). See also llvm.org.
    - `mv ~/Downloads/llvm-project-15.0.7.src ~/Downloads/clang-llvm`
    - `mkdir ~/Downloads/clang-llvm/build`
    - `cd ~/Downloads/clang-llvm/build`
    - `cmake -G Ninja ../llvm -DLLVM_ENABLE_PROJECTS="clang;clang-tools-extra" -DLLVM_BUILD_TESTS=O -DCMAKE_BUILD_TYPE=release -DLLVM_ENABLE_RTTI=ON`
    - **`ninja`**
- build `sourcetrail` with C++ enabled:
    - `cd ~/Downloads/Sourcetrail-master`
    - edit **`CMakeLists.txt`** and around line 186:
        - replace: `find_package(Boost 1.67`   by: `find_package(Boost 1.81`   <<<< SAME VERSION AS BOOST INSTALLED ABOVE
    - `fix` **`CMakeLists.txt`** `for linking as mentioned below` Linkerrors.
    - `rm -rf build/Release    (just in case it's not the first build)`
    - `mkdir -p build/Release`
    - `cd build/Release`
    - `cmake -DCMAKE_BUILD_TYPE="Release" -DClang_DIR=~/Downloads/clang-llvm/build/lib/cmake/clang -DBOOST_ROOT=~/Downloads/boost_1_81_0 -DBUILD_CXX_LANGUAGE_PACKAGE=ON ../..`
        - see `cmake` output example below Sourcetrailcmakeoutput(example).
    - fix `Sourcetrail` source for `llvm 15.0.7` as mentioned below Compileerrors.
    - possibly update `Sourcetrail` version as mentioned below OptionalVersion.
    - **`make`** `(or` **`bear -o sourcetrail.cdb.json make`** `to generate its own Compile Data Base)`
- `Then Sourcetrail is in build/Release/app/`

I don't think the `-DBOOST_ROOT` in the `cmake` command is needed. `cmake` complains the macro is not being used. `Boost` was installed above anyway. So what's important is to specify the right Boost version in `CMakeLists.txt`.

Optional Version

By default, when building like this, the **version** in '`About Sourcetrail`' is `0.0.0`. Not great. Possibly update `build/Release/src/lib_gui/productVersion.h` BEFORE doing `make`. See `VERSION_YEAR`, `VERSION_MINOR` and `VERSION_COMMIT`.

Sourcetrail fixes for `llvm-project-15.0.7`

Compile errors

`Sourcetrail-master/src/lib_cxx/data/parser/cxx/name_resolver/`**`CxxTemplateArgumentNameResolver.cpp`**

```
line 52:      argument.print(pp, os);  --  argument.print(pp, os, true /* IncludeType */);
```

Sourcetrail-master/src/lib_cxx/data/parser/cxx/**PreprocessorCallbacks.h**

```
line 40:      const clang::FileEntry* fileEntry,   --   llvm::Optional< clang::FileEntryRef >
fileEntry,
```

Sourcetrail-master/src/lib_cxx/data/parser/cxx/**PreprocessorCallbacks.cpp**

```
line 61:   const clang::FileEntry* fileEntry,  --   llvm::Optional< clang::FileEntryRef > fileEntry,
line 67:   ---  if (m_currentFileSymbolId && fileEntry.hasValue())
line 69:   ---  const FilePath includedFilePath = m_canonicalFilePathCachegetCanonicalFilePath
(fileEntry.value());
```

Sourcetrail-master/src/lib_cxx/data/parser/cxx/**CanonicalFilePathCache.cpp**:35
Sourcetrail-master/src/lib_cxx/data/parser/cxx/**CanonicalFilePathCache.cpp**:129
Sourcetrail-master/src/lib_cxx/data/parser/cxx/**CxxDiagnosticConsumer.cpp**:86
Sourcetrail-master/src/lib_cxx/data/parser/cxx/**CxxDiagnosticConsumer.cpp**:98
Sourcetrail-master/src/lib_cxx/data/parser/cxx/**utilityClang.cpp**:132

```
if (fileEntry != nullptr && fileEntryisValid())     --- if (fileEntry != nullptr)
```

Link errors

Edit `Sourcetrail-master/`**`CMakeLists.txt`** and around line 411 force `REQ_LLVM_LIBS` list of libraries to link, add this line (after the foreach loop trying to build the `REQ_LLVM_LIBS` list):

```
     set(REQ_LLVM_LIBS "LLVMSupport;LLVMCore;LLVMLibDriver;LLVMPasses;LLVMOption;LLVMAArch64CodeGen;
LLVMAArch64AsmParser;LLVMAMDGPUCodeGen;LLVMAMDGPUAsmParser;LLVMARMCodeGen;LLVMARMAsmParser;
LLVMAVRCodeGen;LLVMAVRAsmParser;LLVMBPFCodeGen;LLVMBPFAsmParser;LLVMHexagonCodeGen;
LLVMHexagonAsmParser;LLVMLanaiCodeGen;LLVMLanaiAsmParser;LLVMMipsCodeGen;LLVMMipsAsmParser;
LLVMMSP430CodeGen;LLVMMSP430AsmParser;LLVMPowerPCCodeGen;LLVMPowerPCAsmParser;LLVMRISCVCodeGen;
LLVMRISCVAsmParser;LLVMSparcCodeGen;LLVMSparcAsmParser;LLVMSystemZCodeGen;LLVMSystemZAsmParser;
LLVMVECodeGen;LLVMVEAsmParser;LLVMWebAssemblyCodeGen;LLVMWebAssemblyAsmParser;LLVMX86CodeGen;
LLVMX86AsmParser;LLVMNVPTXInfo;clangTidyMain;clangDependencyScanning")
```

And run the '`cmake`' + '`make`' commands again.
Probably the right fix is to properly configure `llvm` build (if/when I have time...).

Sourcetrail `cmake` output (example)

```
cmake -DCMAKE_BUILD_TYPE="Release" -DClang_DIR=/home/pierre/Downloads/clang-llvm/build/lib/cmake/clang -
DBOOST_ROOT=~/Downloads/boost_1_81_0 -DBUILD_CXX_LANGUAGE_PACKAGE=ON ../..
-- Version: 0.0.0
-- setting up the buildtype for versioning
-- ccache found
bash: ''
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- 'Treat warnings as errors' policy is disabled
-- Performing Test Terminfo_LINKABLE
-- Performing Test Terminfo_LINKABLE - Success
-- Found Terminfo: /usr/lib/x86_64-linux-gnu/libtinfo.so
-- Found ZLIB: /usr/lib/x86_64-linux-gnu/libz.so (found version "1.2.11")
-- Found LLVM 16.0.0git
-- Trying to find Clang compiler headers in '.' build config in directory '/home/pierre/Downloads/clang-llvm
/build/./lib/clang/16.0.0git/include'.
-- Found headers for '.' build config.
CMake Warning at /usr/share/cmake-3.16/Modules/FindBoost.cmake:1161 (message):
  New Boost version may have incorrect or missing dependencies and imported
  targets
Call Stack (most recent call first):
  /usr/share/cmake-3.16/Modules/FindBoost.cmake:1283 (_Boost_COMPONENT_DEPENDENCIES)
  /usr/share/cmake-3.16/Modules/FindBoost.cmake:1921 (_Boost_MISSING_DEPENDENCIES)
  CMakeLists.txt:188 (find_package)


CMake Warning at /usr/share/cmake-3.16/Modules/FindBoost.cmake:1161 (message):
  New Boost version may have incorrect or missing dependencies and imported
  targets
Call Stack (most recent call first):
  /usr/share/cmake-3.16/Modules/FindBoost.cmake:1283 (_Boost_COMPONENT_DEPENDENCIES)
  /usr/share/cmake-3.16/Modules/FindBoost.cmake:1921 (_Boost_MISSING_DEPENDENCIES)
  CMakeLists.txt:188 (find_package)


CMake Warning at /usr/share/cmake-3.16/Modules/FindBoost.cmake:1161 (message):
  New Boost version may have incorrect or missing dependencies and imported
  targets
Call Stack (most recent call first):
  /usr/share/cmake-3.16/Modules/FindBoost.cmake:1283 (_Boost_COMPONENT_DEPENDENCIES)
  /usr/share/cmake-3.16/Modules/FindBoost.cmake:1921 (_Boost_MISSING_DEPENDENCIES)
  CMakeLists.txt:188 (find_package)


CMake Warning at /usr/share/cmake-3.16/Modules/FindBoost.cmake:1161 (message):
  New Boost version may have incorrect or missing dependencies and imported
  targets
Call Stack (most recent call first):
  /usr/share/cmake-3.16/Modules/FindBoost.cmake:1283 (_Boost_COMPONENT_DEPENDENCIES)
  /usr/share/cmake-3.16/Modules/FindBoost.cmake:1921 (_Boost_MISSING_DEPENDENCIES)
  CMakeLists.txt:188 (find_package)
```

```
-- Found Boost: /home/pierre/Downloads/boost_1_81_0 (found suitable version "1.81.0", minimum required is
"1.81") found components: system program_options filesystem date_time
-- Found Qt 5.12.8
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Building the Java indexer will be skipped. You can enable building this target by setting
'BUILD_JAVA_LANGUAGE_PACKAGE' to 'ON'.
-- Building the Python indexer will be skipped. You can enable building this target by setting
'BUILD_PYTHON_LANGUAGE_PACKAGE' to 'ON'.
-- create symlink: /home/pierre/Downloads/Sourcetrail-master/bin/app/data -> /home/pierre/Downloads
/Sourcetrail-master/build/Release/app/data
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pierre/Downloads/Sourcetrail-master/build/Release
```