

How I Installed SPDROOT In Ubuntu 22.04 LTS OS

Rishav Pandey
rishav160999@gmail.com

05.10.2022

1 Installation of SPDROOT in Ubuntu 22.04 LTS

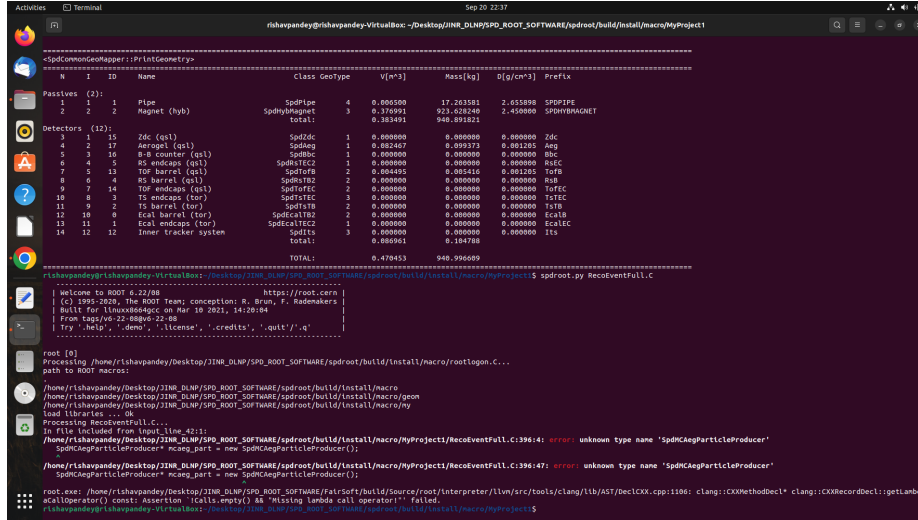
In this article I'm going to talk about how I installed SPDROOT in my ubuntu 22.04 OS. At first, I was exactly following the installation guide present in NICA Gitlab repository. So, during the installation of FairSoft version jun19p2, I faced many errors, like "pythia6 could not be created. Multiple definitions found", "Geant4 could not be created" and many more. However, I resolved almost all of them in sequence but at last there was some error which pushed me install another version of FairSoft. I switched to FairSoft version apr21p2 and fortunately, I was able to install this without any error.

Then, I moved to install FairRoot v18.2.0 which is mentioned in NICA Gitlab repo. I was not able to configure the FairRoot, so I tried to install the same without doing "git checkout v18.2.0". So, I think this installed the latest version of FairRoot in my PC and not the "18.2.0".

After FairSoft and FairRoot was successfully installed, I started my installation of SPDROOT from master branch. Again, during the installation of SPDROOT, I faced configuration error like "Looking for boost, couldn't find boost". So, in this case also I tried to install the same from another branch. I cloned another version of SPDROOT and did git checkout dev-hlit. Fortunately, this version was successfully installed, but during its installation, it complained that your FairRoot has no file named "FairMCTracks.h". So, I searched the same file in FairRoot github repo (under version 18.2.0), and manually added it in my FairRoot in the same path where it was complaining. Now, the manually added file needed to be compiled, so I did some

changes in CMakeLists.txt and LinkDef.h file associated with its path. With this I was done with the installation of SPDROOT.

After all these when I tried to run an example (under macro folder), I was getting errors like “unknown type name ‘SpdMCAegParticleProducer’ ” as shown in fig. 1. So, I tried to search for this file, and found that there are many files missing from my SPDROOT. Then I compared SPDROOT from master branch and SPDROOT from “dev-hlit”. And yes there were many missing files in later, including the one specified above. Manually adding files and compiling again and again could be a bad idea. Even I tried this, but there were many errors.



```

=====
<SpdCommonGeoparser::PrintGeometry>
=====
N  I  ID  Name                                     Class Geotype      V[m3]      Mass[kg]      D[g/cm3]  Prefix
-----
Passives (2):
1  1  1  Pipe                                     SpdPipe           0.000000     17.203581     2.65588    SPDPIPE
2  2  2  Magnet (hyb)                             SpdHybMagnet      0.374991     923.428240     2.45000    SPDHYBMAGNET
Total:
0.383491     946.891821
-----
Detectors (12):
3  1  15  Zdc (qsl)                                SpdZdc            0.000000     0.000000     0.00000    ZDC
4  2  17  Aarmpc (qsl)                             SpdAarmpc         0.000000     0.000000     0.00000    AARMP
5  3  16  B-B counter (qsl)                        SpdBbc            0.000000     0.000000     0.00000    Bbc
6  4  5  RS endcap (qsl)                         SpdRSTEC2        0.000000     0.000000     0.00000    RSTEC
7  5  13  T0F barrel (qsl)                        SpdT0FB           0.004445     0.004416     0.00120    T0FB
8  6  4  RS barrel (qsl)                         SpdRSTEC         0.000000     0.000000     0.00000    RSTEC
9  7  14  T0F endcap (qsl)                        SpdT0FEC         0.000000     0.000000     0.00000    T0FEC
10 8  3  TS barrel (qsl)                         SpdTSTEC         0.000000     0.000000     0.00000    TSTEC
11 9  2  TS barrel (top)                         SpdTSTB          0.000000     0.000000     0.00000    TSTB
12 10 9  Ecal barrel (top)                      SpdEcalTB        0.000000     0.000000     0.00000    EcalTB
13 11 1  Ecal endcap (top)                     SpdEcalTEC2      0.000000     0.000000     0.00000    EcalEC
14 12 12 Inner tracker system              SpdITS            0.000000     0.000000     0.00000    ITS
Total:
0.000061     0.167188
=====
TOTAL:
0.478453     948.996009
=====
Welcome to ROOT 6.32.000
(c) 1995-2020, The ROOT Team; conception: R. Brun, F. Radenmakers
Built for linux64 on Mar 10 2021, 14:26:04
From tags/v6-32-000@22-08
Try 'help', 'demo', 'license', 'credits', 'quit'/'q'

root [0]
Processing /home/rishavpandey/Desktop/12th_DLNP/SPD_ROOT_SOFTWARE/spdroot/build/install/macro/rootlogon.C...
path to ROOT macros:
/home/rishavpandey/Desktop/12th_DLNP/SPD_ROOT_SOFTWARE/spdroot/build/install/macro
/home/rishavpandey/Desktop/12th_DLNP/SPD_ROOT_SOFTWARE/spdroot/build/install/macro/geon
/home/rishavpandey/Desktop/12th_DLNP/SPD_ROOT_SOFTWARE/spdroot/build/install/macro/mg
load libraries ... ok
Processing RecEventFull.C...
In file included from Input_line_421:1:
/home/rishavpandey/Desktop/12th_DLNP/SPD_ROOT_SOFTWARE/spdroot/build/install/macro/MyProject/RecEventFull.C:396:4: error: unknown type name 'SpdMCAegParticleProducer'
  SpdMCAegParticleProducer* mcag_part = new SpdMCAegParticleProducer();
  ~~~~~^
/home/rishavpandey/Desktop/12th_DLNP/SPD_ROOT_SOFTWARE/spdroot/build/install/macro/MyProject/RecEventFull.C:396:47: error: unknown type name 'SpdMCAegParticleProducer'
  SpdMCAegParticleProducer* mcag_part = new SpdMCAegParticleProducer();
  ~~~~~^
root.exe: /home/rishavpandey/Desktop/12th_DLNP/SPD_ROOT_SOFTWARE/rats/soft/build/source/root/interpreter/lin/src/tools/clang/libAST/declCX.c:1106: clang::CXMethodDecl* clang::CXRecordDecl::getLambdaOperator() const: Assertion '!call.empty() && "Missing lambda call operator"' failed:
rshavpandey@rshavpandey-VirtualBox: ~/Desktop/12th_DLNP/SPD_ROOT_SOFTWARE/spdroot/build/install/macro/MyProject$

```

Figure 1: Missing file error

So, I tried a different approach. At first I cloned the SPDROOT from master branch only (because it has all required files, and there will be no error regarding file missing). Then, I configured this SPDROOT with the CMakeLists.txt copied from SPDROOT(dev-hlit). This means that I was using SPDROOT from main branch but CMakeLists.txt was from ‘dev-hlit’ version. I did this because the later was successfully installed (but complained many missing files). Now, with this combination I should have all files with successful installation of SPDROOT. And yes, this combination worked for me. Installation was successful. I was able to run the examples.

But, my work went in vain when I tried to run the analysis script of my project. There was a complain, and it was “The sink does not exist to store persistent branches”.

I tried to look for solution online, but couldn’t find any. However, I noticed that my SPDROOT uses the tree named ‘cbmsim’ instead of ‘spdsim’. So, I found that in my SPDROOT the ‘config’ directory was not built during the installation. The reason was because I was using CMakeLists.txt from another version which uses cbmsim only and hence there was no command mentioned in the CMakeList to build that ‘config’ directory. Hence, in that CMakeList I only added a line “Install(DIRECTORY config DESTINATION .)” [please see fig. 2] and then compiled my SPDROOT again, and this time it built that ‘config’ directory under the ‘install’ directory. So, my SPDROOT started to use ‘spdsim’ but the error “The sink does not exist to store persistent branches” was still there.

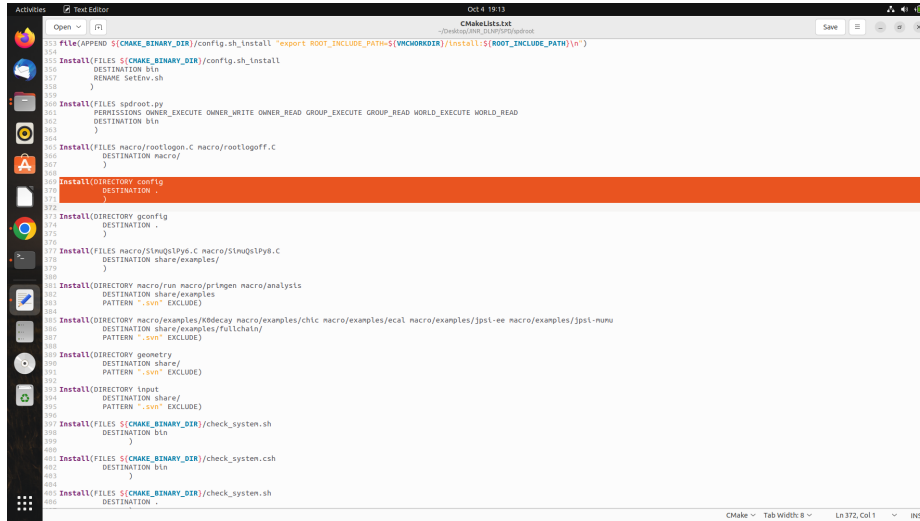


Figure 2: Code to build config directory

I tried to search this string “The sink does not exist to store persistent branches” using ‘grep’ and found that the error was coming from ‘FairRootManager.h’ which was inside FairRoot. Then I compared the FairRootManager.h of my latest version of FairRoot with the FairRootManager.h present in older version which was recommended to install. I found a small change

between those two files. As shown in fig. 3, there is no conditional statement in older version whereas in latest version there is. So, I followed the older version and removed that conditional statement. Compiled FairRoot and SPDROOT followed by installation removed the error. With this my SPDROOT was/is working fine. Also, I completed my project in both the versions of SPDROOT and compared the results/plots obtained. There is almost no difference.

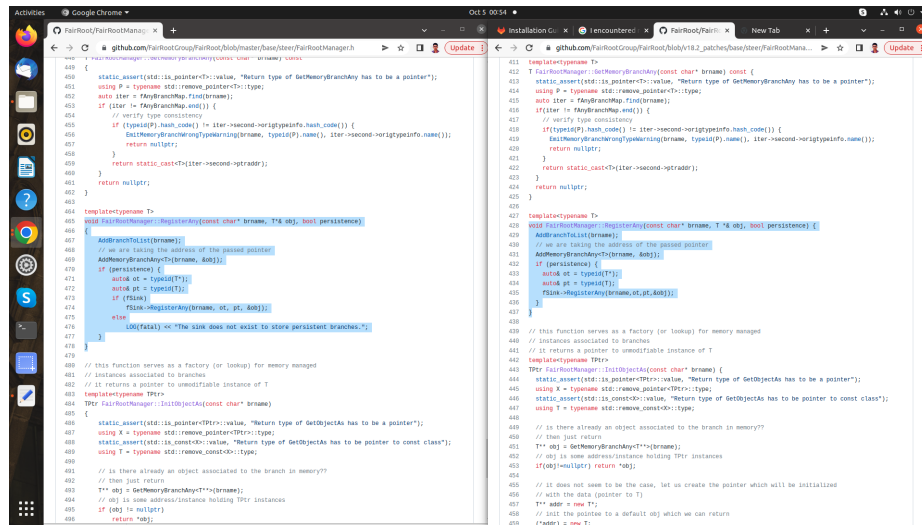


Figure 3: Comparing FairRootManager.h between latest version (shown left) with v18.2.0 (shown right)

But later on I came to know that although my SPDROOT is working fine and producing output, but one of its feature ‘DisplayEvent.C’ doesn’t work and upon running there was an error “no member named ‘SetInputFile’ in ‘SpdRunAna’ [Please see fig. 7]”. On probing I came to know the error was again due to the latest version of FairRoot which I was using and it was due to file named ‘FairRunAna’. I did comparison of this file in both the versions and found that SetInputFile function is missing in latest version which is required to run DisplayEvent.C. I manually added its definition and code in .h and .cxx file respectively as shown in fig. 5. Compiled and installed both FairRoot and SPDROOT again, and the error vanished. It’s working now and I’m happy.

