

Creating root Tuples from LCIO data

The goal is to create a root tuple from data in an LCIO file. This will be done within root using the LCIO root dictionary.

Prerequisite

It is assumed that you already have a working installation of [root](#) and that you have defined the system variables *via*

```
source <rootinstalldir>/bin/thisroot.sh
```

Checkout and build [LCIO](#) including the root dictionary

```
git clone https://github.com/iLCSoft/LCIO.git
cd LCIO
checkout v02-15-02
mkdir build
cd build
cmake -DBUILD_ROOTDICT=ON -DCMAKE_CXX_STANDARD=17 ..
make -j 4 install
cd ..
. setup.sh
cd ..
```

Create a root logon file

This file will be used to load the LCIO root dictionary into root.

Create a file [rootlogon.C](#) with this content:

```
{
gInterpreter->AddIncludePath("$LCIO");
gSystem->Load("${LCIO}/lib/liblcio.so");
gSystem->Load("${LCIO}/lib/liblcioDict.so");
}
```

Write the analysis code

The following code ([writeNTuple.C](#)) provides an example of how to open an LCIO file containing a collection of MCParticle objects and create a tuple containing the momentum and starting position for stable particles from the generator.

```
/******
load LCIO libraries before calling this macro in root:
    gSystem->Load("liblcio");  gSystem->Load("liblcioDict");

******/
/** Example script for using the LCIO root dictionary.
 *  writeNTuple:
 *  reads *.slcio file and creates a simple NTuple from the MCParticle collection.
 *
 */
void writeNTuple(const char* FILEN) {

    std::string rootFileBaseName( FILEN , (strlen(FILEN) ) - strlen(".slcio") ) ;
    std::string rootFileName = rootFileBaseName + std::string(".root") ;
```

```

// std::cout << " rootFileName: " << rootFileName << std::endl ;

std::string mcpName("MCParticle") ;

const char* tupleNames = "px:py:pz:vx:vy:vz:pdg" ;
double px,py,pz,vx,vy,vz ;
int pdg, gs;

//--- create a ROOT file and an NTuple

TFile* file = new TFile( rootFileName.c_str() , "RECREATE");
TNTupleD *tuple= new TNTupleD("MCP", "", tupleNames );
std::cout << " loaded LCIO library and dictionary ... " << std::endl ;
int nEvents = 0 ;
int maxEvt = 10000 ; // change as needed
IO::LCReader* lcReader = IOIMPL::LCFactory::getInstance()->createLCReader() ;
lcReader->open( FILEN ) ;

IMPL::LCCollectionVec emptyCol ;
EVENT::LCEvent* evt = 0 ;
//----- the event loop -----
while( (evt = lcReader->readNextEvent()) != 0  && nEvents < maxEvt ) {

    // UTIL::LCTOOLS::dumpEvent( evt ) ;
    nEvents ++ ;
    // ----- fill the MCParticle collection to tuple

    IMPL::LCCollectionVec* col = (IMPL::LCCollectionVec*) evt->getCollection( mcpName ) ;
    int nMCP = col->getNumberOfElements() ;
    for(int i=0 ; i<nMCP ; ++i){

        EVENT::MCParticle* mcp = (EVENT::MCParticle*) col->getElementAt(i) ;

        if(mcp->getGeneratorStatus() == 1) {

            px = mcp->getMomentum()[0] ;
            py = mcp->getMomentum()[1] ;
            pz = mcp->getMomentum()[2] ;

            vx = mcp->getVertex()[0] ;
            vy = mcp->getVertex()[1] ;
            vz = mcp->getVertex()[2] ;

            pdg = mcp->getPDG() ;

            tuple->Fill(px,py,pz,vx,vy,vz,pgd) ;
        }
    }
}
// ----- end of event loop -----
file->Write() ;
file->Close() ;

delete file ;
std::cout << std::endl
    << " " << nEvents
    << " events read from file: "
    << FILEN << std::endl ;

lcReader->close() ;

delete lcReader ;
}

```

Compile and run the code

```
$root -b
```

```
-----  
| Welcome to ROOT 6.18/04                https://root.cern |  
|                                     (c) 1995-2019, The ROOT Team |  
| Built for linuxx8664gcc on Sep 11 2019, 15:38:23 |  
| From tags/v6-18-04@v6-18-04 |  
| Try '.help', '.demo', '.license', '.credits', '.quit'/'.' |  
-----
```

```
root [0] .L writeNTuple.C
```

```
root [1] writeNTuple("hpsForward_e-_4.5GeV_z-7.5_0_SLIC-v06-00-01_QGSP_BERT_HPS-PhysicsRun2019-v2-4pt5_recon.  
slcio")
```

```
loaded LCIO library and dictionary ...
```

```
10000 events read from file: hpsForward_e-_4.5GeV_z-7.5_0_SLIC-v06-00-01_QGSP_BERT_HPS-PhysicsRun2019-v2-  
4pt5_recon.slcio
```

```
root [2] .q
```