

TPC Offline Manual

P. Christiansen, H. Helstrup, M. Ivanov, M. Kowalski, and J. Wiechula

v0.1, January 31, 2008

Abstract

This document is meant to give an overview of TPC offline code relevant for expert shifters and for people interested in analyzing TPC data.

Contents

1	Environment setup on lxplus.cern.ch	2
2	Data Location and Access	2
2.1	Raw data	2
2.1.1	Access to raw data	3
2.2	Reconstructed data	3
2.2.1	Analysis of reconstructed data	3
3	Data Monitors	4
3.1	The TPC Pad Monitor	4
3.2	AliEve	4
4	Calibration, Shuttle and QA	5
4.1	Calibration Classes	5
4.1.1	AliTPCdataQA	5
4.2	The Shuttle	5
4.3	TPC QA	5
5	Simulation and Reconstruction	7
5.1	Simulation	7
5.1.1	TPC versions	8
5.2	Reconstruction	9
5.2.1	Clustering	9
5.2.2	Tracking	9
5.3	Reconstruction of raw data	9

1 Environment setup on lxplus.cern.ch

This part is valid only for login on lxplus.cern.ch. In case of problems contact marian.ivanov@cern.ch.

Alien setup:

- Use bash shell: *bash*
- *source /afs/cern.ch/user/m/miranov/public/.balien*
- *alien-token-init \$USERNAME*
- Default message example - Your token is valid until: Sat Feb 9 15:21:17 2008
- In case of problem be sure that your certificate is fine. e.g. Check the presence of the .globus director: *ls /globus*
- *source /tmp/gclient.env.\$UID*

AliRoot setup:

- The central - default aliroot setup - type *new*
- The TPC setup - regularly updated HEAD version of HEAD. Use this setup if you want to use TPC specific features. The will be
source /afs/cern.ch/user/m/miranov/public/.balice

2 Data Location and Access

2.1 Raw data

The naming scheme for the raw files is:

<year2digits><runno9digits><gdc3digits>.<fileno>0.root

Example: 0700001506700410.root

A single tag file is created for each GDC.

The directories depends on whether the run was recorded in standalone or in global mode. Unfortunately it is not always possible to see from the logbook overview if a run was started as a stand alone or a global run (DAQ is always standalone).

Standalone runs:

CASTOR: */castor/cern.ch/alice/tpc/<year>/<month>/<day>/<hour>*

AliEn: */alice/data/<year>/LHC<year2digits><quarter>.TPC/<runno>/raw/*

Where `<quarter>` can be `w` (winter).

Global runs:

CASTOR: `/castor/cern.ch/alice/data/<year>/<month>/<day>/<hour>`

AliEn: `/alice/data/<year>/LHC<year2digits><quarter>/<runno>/raw/`

2.1.1 Access to raw data

The files located on CASTOR can be accessed directly from lxplus machines where one can use `nsls` to list files and `rfcp` to copy files to local disk. The files can also be accessed directly inside a ROOT session if ROOT is compiled with `rfio` option (`--enable-rfio`).

Example of ROOT filename:

`rfio:/castor/cern.ch/alice/tpc/2007/12/13/05/07000011524011.6380.root`

Nb! You have to specify `rfio:` to let ROOT know that it is a remote/CASTOR file you wish to open.

The data registered in AliEn can be directly streamed in ROOT (if ROOT is compiled with AliEn support). One has to get an AliEn token (`alien-token-init`) and then do

`Tgrid::Connect("alien://");`

inside root, aliroot or alieve.

Example of ROOT path-name:

`alien:///alice/data/2007/LHC07w/TPC/000010897/raw/07000010897011.20.root`

NB! `alien://` tells ROOT that it is an AliEn file you wish to open.

2.2 Reconstructed data

The automatically reconstructed data can be found on AliEn. Instead of the raw directory, the files are stored under ESDs.

Example of directory with reconstructed files:

`/alice/data/2007/LHC07w/000014493/ESDs/00120/`

The name of the directory 00120 corresponds to the last 5 digits in the raw file name.

2.2.1 Analysis of reconstructed data

An example `AnalysisTask` for looping over the clusters in reconstructed tracks is under development.

An early working example can be found at:

http://www.hep.lu.se/staff/christiansen/tpc_ana_track_example.tgz

3 Data Monitors

There two types of monitors. The TPC pad monitor and the standard AliEve monitors.

3.1 The TPC Pad Monitor

The TPC pad monitor is maintained by Stefan Kniese as both an AliRoot and standalone application. The standalone application is used for installation on the TPC shift machine in the counting room.

The AliRoot version can be run from inside an aliroot session like this:

```
.L $ALICE_ROOT/lib/tgt_linux/libTPCmon.so
.L $ALICE_ROOT/TPC/macros/TPCMonitor.C
TPCMonitor()
```

Now the GUI starts and you should click:

Sel.Format - Select ROOT and remember to press Select Entry

Sel.File - Open file - can soon be alien file

Press Next Event until the data is loaded

3.2 AliEve

AliEve is part of AliRoot and is a general purpose 3d graphical monitoring framework. It is called with the command `alieve`. The monitoring display for different detectors is controlled by macros. For the TPC, AliEve can be used to display both raw data and reconstructed data.

To run the TPC raw display one has to call the following macro inside `alieve`:

```
.L ../test-macros/tpc_gui.C
tpc_gui(<filename>, <event_no>)
```

In the GUI you can double click on `TPCLoader` to control the many different draw options. Click `Help` to see how to navigate in the opengl window. You can also get the menu of graphical elements by clicking on them (`SHIFT+right click`) and see 1d projections of the raw data (`CTRL+left click`).

4 Calibration, Shuttle and QA

4.1 Calibration Classes

4.1.1 AliTPCdataQA

This special calibration class is used for QA of raw data, but can also be run standalone. It is currently under development. The problem is how to extract relevant parameters from the raw data which are not dominated by the large noise.

4.2 The Shuttle

4.3 TPC QA

A first version of the TPC QA have recently been committed. It contains the basic structure and a few histograms.

The TPC QA is made in the standard ALICE QA framework and contains the 3 classes: AliTPCQAChecker, AliTPCQADataMakerSim, AliTPCQADataMakerRec.

The DataMaker classes makes the histograms from the relevant data — hits and digits for Sim, and raw, rec points and ESDs for Rec — and the Checker contains the rules for comparing the histograms to the reference histograms (which currently does not exist). The Checker is currently empty which means that the default comparison is employed.

The implementation is slightly different than for the other QA classes since we have chosen to have histogram pointers in the class to make the class easier to debug. The only place where this could cause a problem is in the copy constructor where the histogram pointers have to be assigned after the “lists” have been copied. This has been implemented, but not tested.

In the QA Checker for raw data we use the more powerful TPC calibration framework (the class AliTPCdataQA, see above) so that the method for handling the raw data QA is non-standard. Instead of filling the histograms in the MakeRaws method, the calibration object is filled, and it is first in the method EndOfDetectorCycle when we have called AliTPCdataQA::Analyse that the histogram is projected from the calibration object and added to the list.

The QA can be run like this inside aliroot:

```
AliQADataMakerSteer* dataMaker = new AliQADataMakerSteer();
dataMaker->Run("TPC", <type>, filename);
```

Where filename does not have to be supplied for data in the same directory (except for raw data), and <type> can be:

```
AliQA::kRAWS
AliQA::kHITS
AliQA::kSDIGITS
```

AliQA::kDIGITS
AliQA::kRECPPOINTS
AliQA::kESDS

5 Simulation and Reconstruction

This section gives a short overview of the TPC simulation and reconstruction software. Other sources of information can be found in the ALICE TPC TDR [1], the ALICE Physics Performance Reports [3, 4] (1: p.1620-1627, updated description of TPC, 2: p. 1326-1336, description of clustering, tracking algorithms and performance).

The TPC simulation and reconstruction consists of the following steps:

- Energy loss simulation. The physics routine which is called by e.g. GEANT3. Output: TPC.Hits.root.
- Signal generation. The electrons drift to the ROCs and are amplified and digitized. Output: TPC.Digits.root.
- Clustering. The first step of reconstruction is the grouping of signals into clusters. The same algorithm is applied to real data and simulated data. Output: TPC.RecPoints.root.
- Tracking. The last step of reconstruction is the grouping of clusters into tracks. The same algorithm is applied to real data and simulated data. Output: AliESDs.root.

This section gives a short overview of the TPC simulation and reconstruction software. Other sources of information can be found in the ALICE TPC TDR [1], the ALICE Physics Performance Reports [3, 4] (1: p.1620-1627, updated description of TPC, 2: p. 1326-1336, description of clustering, tracking algorithms and performance).

The TPC simulation and reconstruction consists of the following steps:

- Energy loss simulation. The physics routine which is called by e.g. GEANT3. Output: TPC.Hits.root.
- Signal generation. The electrons drift to the ROCs and are amplified and digitized. Output: TPC.Digits.root.
- Clustering. The first step of reconstruction is the grouping of signals into clusters. The same algorithm is applied to real data and simulated data. Output: TPC.RecPoints.root.
- Tracking. The last step of reconstruction is the grouping of clusters into tracks. The same algorithm is applied to real data and simulated data. Output: AliESDs.root.

5.1 Simulation

The simulation is currently working for GEANT3 and FLUKA simulations.

How to switch between GEANT3 and FLUKA? In Config*.C macro one has to:

- replace


```
// libraries required by geant321
// if defined(__CINT__)
gSystem->Load("libgeant321");
// endif

new TGeant3TGeo("C++ Interface to Geant3");

with
gSystem->Load("libTFluka");
new TFluka("C++ Interface to Fluka",0/*verbositylevel*/);
```
- add a line `TPC->SetPrimaryIonisation();`

5.1.1 TPC versions

We have 4 TPC versions:

- AliTPCv0 - no sensitive volumes, geometry as in other versions
- AliTPCv1 - fast simulation - the sensitive elements are thin gas strips placed at the pad center, where space-points are created, then the coordinates are smeared according the space-point resolution
- AliTPCv2 - the default version - detailed simulation
In the case of GEANT3 the distance to the next primary ionization is calculated in `AliTPCv2::StepManager` method. In case of FLUKA it is handled by the FLUKA code itself. The energy loss simulation is well described in the TPC TDR [1].
- AliTPCv3 - used only for the space-charge studies, the sensitive volume is the entire drift gas.
- AliTPCv4 - used for Krypton simulations.

The comparison of the energy loss simulation with test beam data is covered in the note [5]. The result of the simulation are hits - the space point coordinates and the track label. Hits from the simulation are converted to Digits in:

`AliTPC::Hits2Digits()` which runs in a loop over sectors. the relevant methods are:

- `MakeSector` (drift and amplification) - each electron is transported to the readout chamber and amplified.
- `DigitizeRow` - for each electron at the given row the signal is calculated in the method `GetSignal`. This method returns the track label. For every digit labels of 3 tracks contributing mostly to this digit are stored as well. This is controlled by the `GetList` method.

In time direction a Gaussian response is assumed in pad direction we use the Gatti-Mathieson function [2].

It turned out recently that the response function after filtering and baseline restoration is asymmetric and one can eventually change this.

5.2 Reconstruction

Parameters used by the reconstruction are set in the `AliTPCRecoParam`, e.g., the `Ct-gRange`, which removes clusters with angles less than 45 degrees (default) with respect to the beam axis, where the track density is too high in Pb+Pb collisions, but which are important in cosmic ray data.

If one wants to have access to the TPC cluster information in the reconstructed tracks it is important to call

```
AliTPCReconstructor::SetStreamLevel(1) and
```

```
reco.SetWriteESDfriend(kTRUE)
```

before calling the reconstruction.

5.2.1 Clustering

For each row in a sector the signals are stored in an array of size $N_{\text{timebins}} \times N_{\text{pads}}$. In the code there is a loop over this matrix and for each peak (local maximum) one makes a cluster.

The clusters are matrices of 5×5 (pads \times timebins) and the position in space and time is calculated from a charge weighted average over this matrix. The total charge Q is the sum of all charge and Q_{MAX} is the charge at the center of cluster (maximum).

For small clusters a symmetrization algorithm is applied to the cluster which has been discussed if it should be removed since it introduces systematic binning effects.

5.2.2 Tracking

5.3 Reconstruction of raw data

References

- [1] ALICE Collaboration, “Time Projection Chamber”, ALICE TDR 7, CERN/LHCC 2000-001.
- [2] J.S. Gordon and E. Mathieson, Nucl. Instrum. Methods 227 (1984) 267;
E. Mathieson and J.S. Gordon, Nucl. Instrum. Methods 227 (1984) 277;
J.R. Thompson et al., Nucl. Instrum. Methods A234 (1985) 505;
E. Mathieson, Nucl. Instrum. Methods A270 (1988) 602.
- [3] F. Carminati *et al.* [ALICE Collaboration], J. Phys. G 30, 1517 (2004).

- [4] B. Alessandro *et al.* [ALICE Collaboration], J. Phys. G **32**, 1295 (2006).
- [5] P. Christiansen and P. Gros, for the Alice TPC Collaboration, “Benchmarking the AliRoot TPC simulation against the 2004 TPC test beam data”, draft for ALICE note.