

# Table of Contents

<b>7.3.1 Offline Primary Vertex Finding</b> .....	<b>1</b>
Contents.....	1
Primary Vertex reconstruction.....	1
Vertex reconstruction analysis.....	1
Review status.....	2

## 7.3.1 Offline Primary Vertex Finding

Complete: 

Detailed Review status

### Contents

- Primary Vertex reconstruction
- Vertex reconstruction analysis
- Review status

### Primary Vertex reconstruction

The reconstruction of Primary vertices using the full tracks is part of the default reconstruction sequence. It is done using the `generalTracks` track collection. The vertices are stored in the event as `reco::Vertex` collections. Two collections are stored in the event:

- `offlinePrimaryVertices` : The standard vertex collection, where the `offlineBeamSpot` is used to filter the tracks. This is the collection which we advise to use.
- `offlinePrimaryVerticesWithBS` : For this collection, the `offlineBeamSpot` is used not only to filter the tracks, but also as an additional constraint in the fit. It improves the resolution of the vertices, but we do not yet advise to use it until further robustness studies are done, especially for the initial conditions.

Opening a data file in root, the data of the primary vertices (such as the coordinates, chi-squared, degrees of freedom) can be viewed directly. More information on how to redo the primary vertex reconstruction is given in the offline guide.

From CMSSW 1.8.0 onwards, if no reconstructed vertex is found in an event using the tracks, a vertex based on the `[SWGuideFindingBeamSpot][beam-spot]` is put into the event. In this case, the vertex contains no tracks (as none have been used), the  $\chi^2$  and the ndof are 0, and the flag `isFake()` is set to true.

### Vertex reconstruction analysis

In cmsRun mode, the primary vertices can be analyzed with the `CMS.PrimaryVertexAnalyzer` module by running `demoAnalyzePrimaryVertex_cfg.py`:

```
cmsRun demoAnalyzePrimaryVertex_cfg.py
```

in the same directory. A root histogram file `simpleVertexAnalyzer.root` is produced, that, in addition to histograms of the fitted position, covariance,  $\chi^2$  and degrees of freedom, also contains results of sanity checks of the number of valid track links and invalid fit parameters.

```
root simpleVertexAnalyzer.root
pullx->Draw()
eff->Draw()
nans->Draw()
```

A simple test of the input track parameters is done by

```
cmsRun demoAnalyzeTracks_cfg.py
```

which produces histograms of track parameter pulls. For instance, to look at curvature and z0:

```
root simpleTrackAnalyzer.root  
pull10->Draw()  
pull14->Draw()
```

## Review status

Reviewer/Editor and Date (copy from screen)	Comments
Main.werdmann - 04 Dec 2006	update
ThomasSpeer - 11 Feb 2008	update
ThomasSpeer - 16 Sep 2008	update for 2.1.X

Responsible: ThomasSpeer

Last reviewed by: ThomasSpeer - 16 Sep 2008

---

This topic: CMSPublic > WorkBookOfflinePrimaryVertexFinding

Topic revision: r42 - 2010-08-06 - SudhirMalik



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback