

# Documentation with Doxygen

V. Kireyeu

**Doxygen** is a tool for **generating documentation** from **annotated sources** (C++ and some others):

- On-line documentation in HTML and/or an off-line reference manual via LaTeX – can be then generated a PDF document from a set of documented source files. The documentation is extracted directly from the sources – consistent with the source code.
- Extract the code structure from undocumented source files, visualize the relations between the various elements by means of include dependency graphs, inheritance diagrams, and collaboration diagrams, which are all generated automatically.
- Create normal documentation (like the official doxygen user manual).

Doxygen is developed under Mac OS X and Linux. Executables for Windows are also available.

<https://www.doxygen.nl/>

# Configuration

doxygen -g <config-file> – will generate a template configuration file

```
PROJECT_NAME = MpNuclei  
...  
JAVADOC_AUTOBRIEF      = YES  
...  
EXTRACT_PRIVATE         = YES  
EXTRACT_PRIV_VIRTUAL   = YES  
EXTRACT_PACKAGE         = YES  
EXTRACT_STATIC          = YES  
EXTRACT_LOCAL_CLASSES   = YES  
...  
HIDE_UNDOC_MEMBERS     = YES  
RECURSIVE              = YES  
...  
GENERATE_HTML           = YES  
GENERATE_LATEX          = YES
```

**Generate documentation:**

doxygen <config-file>

To also make PDF:

cd latex  
make

All possible parameters are described in the generated template file or here:  
<https://www.doxygen.nl/manual/config.html>

# Configuration

## **EXTRACT\_PRIVATE**

If the EXTRACT\_PRIVATE tag is set to YES, all private members of a class will be included in the documentation.

The default value is: NO.

## **EXTRACT\_PRIV\_VIRTUAL**

If the EXTRACT\_PRIV\_VIRTUAL tag is set to YES, documented private virtual methods of a class will be included in the documentation.

The default value is: NO.

## **EXTRACT\_PACKAGE**

If the EXTRACT\_PACKAGE tag is set to YES, all members with package or internal scope will be included in the documentation.

The default value is: NO.

## **EXTRACT\_STATIC**

If the EXTRACT\_STATIC tag is set to YES, all static members of a file will be included in the documentation.

The default value is: NO.

All possible parameters are described in the generated template file or here:  
<https://www.doxygen.nl/manual/config.html>

# Markdown support

```
USE_MDFILE_AS_MAINPAGE = README.md
```

The README.md will be placed on the main page (index.html) – reuse of the README.md file used on GitLab/GitHub.

README.md

```
## Dependencies {#dependencies}
### Branches {#branches}
- MCTrack
- TpcKalmanTrack
- ZdcDigi
- Vertex
- MPDEvent
- TOFMatching

### Wagons {#wagons}
- evCentrality
- evPID

## Usage {#usage}
Add these lines to your "train" macro (e.g. `RunAnalyses.C`):
```c
MpdNuclei taskNuclei("taskNuclei", "taskNuclei", "NucleiAna.json");
man.AddTask(&taskNuclei);
```

```



Documentation main page

## Dependencies

### Branches

- MCTrack
- TpcKalmanTrack
- ZdcDigi
- Vertex
- MPDEvent
- TOFMatching

### Wagons

- evCentrality
- evPID

### Usage

Add these lines to your "train" macro (e.g. RunAnalyses.C):

```
MpdNuclei taskNuclei("taskNuclei", "taskNuclei", "NucleiAna.json");
man.AddTask(&taskNuclei);
```

# Special comment blocks

- **Javadoc style**

```
/**  
 * ... text ...  
 */
```

- **Qt style**

```
/*!  
 * ... text ...  
 */
```

```
/*!  
 ... text ...  
 */
```

- **(at least) two C++ comment lines + an additional slash or an exclamation mark**

```
///  
/// ... text ...  
///
```

```
///!  
///!... text ...  
///!
```

- **Custom**

```
*****  
* ... text  
*****
```

```
|||||||  
/// ... text ...  
|||||||
```



ROOT uses this style

# Special comment blocks

## Documentation after members

```
int var; /*!< Detailed description after the member
*/
int var; /**< Detailed description after the member
*/
int var; //!< Detailed description after the member
//!<
int var; ///< Detailed description after the member
///<
int var; //!< Brief description after the member
int var; ///< Brief description after the member
```

## Member groups

```
/** @name Group1
 * Description of group 1.
 */
///@{
/** Function 2 in group 1. Details. */
void Memgrp_Test::func2InGroup2() {}
/** Function 1 in group 1. Details. */
void Memgrp_Test::func1InGroup2() {}
///@}
```

## Lists

```
/**
 * Text before the list
 * - list item 1
 *   - sub item 1
 *     - sub sub item 1
 *
 *       .
 *       The dot above ends the sub sub item list.
 *
 * - list item 2
 *
 *
 * More text in the same paragraph.
 */
```

# Examples

```
/** This subroutine performs the track quality checks.

Return values are:
- true (1) if checks are not passed ("bad" track -- skip);
- false (0) if checks are passed ("good" track -- analyse);

The track can be skipped if:
- The number of hits is less than MpdNuclei::s_tr_NHits
- The n-sigma for DCAx is larger than MpdNuclei::s_tr_NSigmaDCAx
- The n-sigma for DCAy is larger than MpdNuclei::s_tr_NSigmaDCAy
- The n-sigma for DCAz is larger than MpdNuclei::s_tr_NSigmaDCAz
- The transverse momentum is less than MpdNuclei::s_tr_LowPtCut

\param track MpdTrack to analyse
*/
bool MpdNuclei::bad_track(MpdTrack* track){
    if (track->GetNofHits() <= s_tr_NHits)                      return true; // Number of hits cut
    if (fabs(track->GetNSigmaDCAx()) > s_tr_NSigmaDCAx)       return true; // |DCAx| cut
    if (fabs(track->GetNSigmaDCAy()) > s_tr_NSigmaDCAy)       return true; // |DCAy| cut
    if (fabs(track->GetNSigmaDCAz()) > s_tr_NSigmaDCAz)       return true; // |DCAz| cut
    if (fabs(track->GetPt()) <= s_tr_LowPtCut)                 return true; // Low transverse momentum cut
    return false;
}
```

# Examples

Doxygen documentation block

```
/** This subroutine performs the track quality checks.

Return values are:
- true (1) if checks are not passed ("bad" track -- skip);
- false (0) if checks are passed ("good" track -- analyse);

The track can be skipped if:
- The number of hits is less than MpdNuclei::s_tr_NHits
- The n-sigma for DCAx is larger than MpdNuclei::s_tr_NSigmaDCAx
- The n-sigma for DCAy is larger than MpdNuclei::s_tr_NSigmaDCAy
- The n-sigma for DCAz is larger than MpdNuclei::s_tr_NSigmaDCAz
- The transverse momentum is less than MpdNuclei::s_tr_LowPtCut

\param track MpdTrack to analyse
*/
bool MpdNuclei::bad_track(MpdTrack* track){
    if (track->GetNofHits() <= s_tr_NHits)                      return true; // Number of hits cut
    if (fabs(track->GetNSigmaDCAx()) > s_tr_NSigmaDCAx)        return true; // |DCAx| cut
    if (fabs(track->GetNSigmaDCAy()) > s_tr_NSigmaDCAy)        return true; // |DCAy| cut
    if (fabs(track->GetNSigmaDCAz()) > s_tr_NSigmaDCAz)        return true; // |DCAz| cut
    if (fabs(track->GetPt()) <= s_tr_LowPtCut)                  return true; // Low transverse momentum cut
    return false;
}
```

# Examples

Doxygen documentation block

```
/** This subroutine performs the track quality checks.  
  
Return values are:  
- true (1) if checks are not passed ("bad" track -- skip);  
- false (0) if checks are passed ("good" track -- analyse);
```

The track can be skipped if:

- The number of hits is less than MpdNuclei::s\_tr\_NHits
- The n-sigma for DCAx is larger than MpdNuclei::s\_tr\_NSigmaDCAx
- The n-sigma for DCAy is larger than MpdNuclei::s\_tr\_NSigmaDCAy
- The n-sigma for DCAz is larger than MpdNuclei::s\_tr\_NSigmaDCAz
- The transverse momentum is less than MpdNuclei::s\_tr\_LowPtCut

\param track MpdTrack to analyse

\*/

```
bool MpdNuclei::bad_track(MpdTrack* track){  
    if (track->GetNofHits() <= s_tr_NHits) return true; // Number of hits cut  
    if (fabs(track->GetNSigmaDCAx()) > s_tr_NSigmaDCAx) return true; // |DCAx| cut  
    if (fabs(track->GetNSigmaDCAy()) > s_tr_NSigmaDCAy) return true; // |DCAy| cut  
    if (fabs(track->GetNSigmaDCAz()) > s_tr_NSigmaDCAz) return true; // |DCAz| cut  
    if (fabs(track->GetPt()) <= s_tr_LowPtCut) return true; // Low transverse momentum cut  
    return false;  
}
```



Normal C++ style comments  
(not processed)

# Examples

Brief description



```
/** This subroutine performs the track quality checks.
```

Return values are:

- true (1) if checks are not passed ("bad" track -- skip);
- false (0) if checks are passed ("good" track -- analyse);

The track can be skipped if:

- The number of hits is less than MpdNuclei::s\_tr\_NHits
- The n-sigma for DCAx is larger than MpdNuclei::s\_tr\_NSigmaDCAx
- The n-sigma for DCAy is larger than MpdNuclei::s\_tr\_NSigmaDCAy
- The n-sigma for DCAz is larger than MpdNuclei::s\_tr\_NSigmaDCAz
- The transverse momentum is less than MpdNuclei::s\_tr\_LowPtCut

```
\param track MpdTrack to analyse
```

```
*/
```

```
bool MpdNuclei::bad_track(MpdTrack* track){  
    if (track->GetNofHits() <= s_tr_NHits) return true; // Number of hits cut  
    if (fabs(track->GetNSigmaDCAx()) > s_tr_NSigmaDCAx) return true; // |DCAx| cut  
    if (fabs(track->GetNSigmaDCAy()) > s_tr_NSigmaDCAy) return true; // |DCAy| cut  
    if (fabs(track->GetNSigmaDCAz()) > s_tr_NSigmaDCAz) return true; // |DCAz| cut  
    if (fabs(track->GetPt()) <= s_tr_LowPtCut) return true; // Low transverse momentum cut  
    return false;  
}
```

Doxygen documentation block



Normal C++ style comments  
(not processed)

# Examples

Brief description



```
/** This subroutine performs the track quality checks.
```

Return values are:

- true (1) if checks are not passed ("bad" track -- skip);
- false (0) if checks are passed ("good" track -- analyze);

The track can be skipped if:

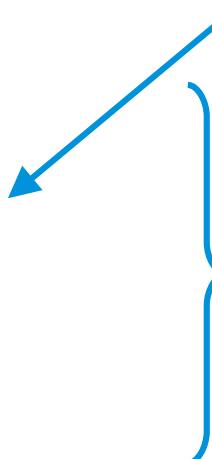
- The number of hits is less than MpdNuclei::s\_tr\_NHits
- The n-sigma for DCAx is larger than MpdNuclei::s\_tr\_NSigmaDCAx
- The n-sigma for DCAy is larger than MpdNuclei::s\_tr\_NSigmaDCAy
- The n-sigma for DCAz is larger than MpdNuclei::s\_tr\_NSigmaDCAz
- The transverse momentum is less than MpdNuclei::s\_tr\_LowPtCut

```
\param track MpdTrack to analyse
```

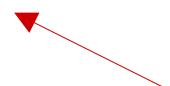
```
*/
```

```
bool MpdNuclei::bad_track(MpdTrack* track){  
    if (track->GetNofHits() <= s_tr_NHits)                      return true; // Number of hits cut  
    if (fabs(track->GetNSigmaDCAx()) > s_tr_NSigmaDCAx)        return true; // |DCAx| cut  
    if (fabs(track->GetNSigmaDCAy()) > s_tr_NSigmaDCAy)        return true; // |DCAy| cut  
    if (fabs(track->GetNSigmaDCAz()) > s_tr_NSigmaDCAz)        return true; // |DCAz| cut  
    if (fabs(track->GetPt()) <= s_tr_LowPtCut)                  return true; // Low transverse momentum cut  
    return false;  
}
```

Doxxygen documentation block



Full description



Normal C++ style comments  
(not processed)

# Examples

Brief description



```
/** This subroutine performs the track quality checks.
```

Return values are:

- true (1) if checks are not passed ("bad" track -- skip);
- false (0) if checks are passed ("good" track -- analyze);

The track can be skipped if:

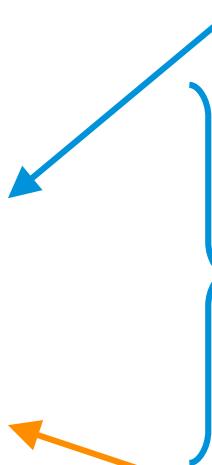
- The number of hits is less than MpdNuclei::s\_tr\_NHits
- The n-sigma for DCAX is larger than MpdNuclei::s\_tr\_NSigmaDCAX
- The n-sigma for DCAY is larger than MpdNuclei::s\_tr\_NSigmaDCAY
- The n-sigma for DCAz is larger than MpdNuclei::s\_tr\_NSigmaDCAz
- The transverse momentum is less than MpdNuclei::s\_tr\_LowPtCut

```
\param track MpdTrack to analyse
```

```
*/  
bool MpdNuclei::bad_track(MpdTrack* track){  
    if (track->GetNofHits() <= s_tr_NHits) return true; // Number of hits cut  
    if (fabs(track->GetNSigmaDCAx()) > s_tr_NSigmaDCAX) return true; // |DCAx| cut  
    if (fabs(track->GetNSigmaDCAY()) > s_tr_NSigmaDCAY) return true; // |DCAY| cut  
    if (fabs(track->GetNSigmaDCAz()) > s_tr_NSigmaDCAz) return true; // |DCAz| cut  
    if (fabs(track->GetPt()) <= s_tr_LowPtCut) return true; // Low transverse momentum cut  
    return false;  
}
```

Doxxygen documentation block

Full description



Reference to the class member  
will be created automatically



Normal C++ style comments  
(not processed)

# Examples

## Brief description



/\*\* This subroutine performs the track quality checks.

Return values are:

- true (1) if checks are not passed ("bad" track -- skip);
- false (0) if checks are passed ("good" track -- analyze);

The track can be skipped if:

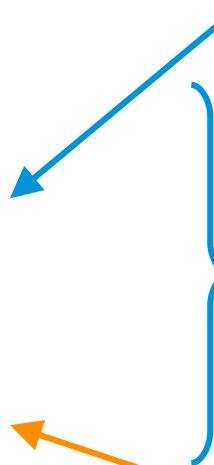
- The number of hits is less than MpdNuclei::s\_tr\_NHits
- The n-sigma for DCAX is larger than MpdNuclei::s\_tr\_NSigmaDCAX
- The n-sigma for DCAY is larger than MpdNuclei::s\_tr\_NSigmaDCAY
- The n-sigma for DCAz is larger than MpdNuclei::s\_tr\_NSigmaDCAz
- The transverse momentum is less than MpdNuclei::s\_tr\_LowPtCut

\param track MpdTrack to analyse

```
*/  
bool MpdNuclei::bad_track(MpdTrack* track){  
    if (track->GetNofHits() <= s_tr_NHits) return true; // Number of hits cut  
    if (fabs(track->GetNSigmaDCAx()) > s_tr_NSigmaDCAX) return true; // |DCAx| cut  
    if (fabs(track->GetNSigmaDCAY()) > s_tr_NSigmaDCAY) return true; // |DCAY| cut  
    if (fabs(track->GetNSigmaDCAz()) > s_tr_NSigmaDCAz) return true; // |DCAz| cut  
    if (fabs(track->GetPt()) <= s_tr_LowPtCut) return true; // Low transverse momentum cut  
    return false;  
}
```

## Doxygen documentation block

### Full description



Reference to the class member  
will be created automatically

For functions one can use the `@param` command to document the parameters and then use `[in]`, `[out]`, `[in,out]` to document the direction.

Normal C++ style comments  
(not processed)

# Examples

## ◆ **bad\_track()**

```
bool MpdBoson::bad_track ( MpdBoson * track )
```

This subroutine performs the track quality checks.

Return values are:

- true (1) if checks are not passed ("bad" track – skip);
- false (0) if checks are passed ("good" track – analyze);

The track can be skipped if:

- The number of hits is less than [MpdBoson::s\\_tr\\_NHits](#)
- The n-sigma for DCAx is larger than [MpdBoson::s\\_tr\\_NSigmaDCAx](#)
- The n-sigma for DCAy is larger than [MpdBoson::s\\_tr\\_NSigmaDCAy](#)
- The n-sigma for DCAz is larger than [MpdBoson::s\\_tr\\_NSigmaDCAz](#)
- The transverse momentum is less than [MpdBoson::s\\_tr\\_LowPtCut](#)

### Parameters

**track** MpdBoson to analyse

## Private Member Functions

bool **bad\_event** (MpdBosonEvent &event)

This subroutine performs the event quality checks.

bool **bad\_track** (MpdBoson \*track)

This subroutine performs the track quality checks.

Definition at line 309 of file [MpdBoson.cxx](#).

# Examples

```
/** @name TPC dE/dx histograms */
///@{
    TH2F *h__dedx
    std::vector<TH2F*>
centrality bins
    std::vector<TH2F*>
    std::vector<TH2F*>
(N-Sigma method)
    std::vector<TH2F*>
///@}

/** @name Identified particles phase-space histograms
 */
///@{
    std::vector<std::vector<TH2F*>>
PDG code in selected centrality bins
    std::vector<std::vector<TH2F*>>
wagon (N-Sigma method) in selected centrality bins
    std::vector<std::vector<TH2F*>>
///@}

    = nullptr; //;< Common dE/dx histogram for all particle species
    hv__dedx; //;< Common dE/dx histograms for all particle species in selected
    hv__dedx_mc; //;< dE/dx histograms for particles identified by the MC PDG code
    hv__dedx_evpid; //;< dE/dx histograms for particles identified by the evPID wagon
    hv__dedx_parbb;
```

# Examples

## TPC dE/dx histograms

`TH2F * h_dedx = nullptr`

Common dE/dx histogram for all particle species.

`std::vector< TH2F * > hv_dedx`

Common dE/dx histograms for all particle species in selected centrality bins.

`std::vector< TH2F * > hv_dedx_mc`

dE/dx histograms for particles identified by the MC PDG code

`std::vector< TH2F * > hv_dedx_evpid`

dE/dx histograms for particles identified by the evPID wagon (N-Sigma method)

## Identified particles phase-space histograms

`std::vector< std::vector< TH2F * > > hv_pteta_mc`

Phase-space histograms for particles identified by the MC PDG code in selected centrality bins.

`std::vector< std::vector< TH2F * > > hv_pteta_evpid`

Phase-space histograms for particles identified by the evPID wagon (N-Sigma method) in selected centrality bins.

# Examples

```
/*! \file postprocess_nuclei.C
\brief Postprocessing macro for the MpRoot "nuclei" wagon.

New subdirectories will be created to store the produced histograms (PDF):
- plots/efficiency/tpc
- plots/efficiency/tof
- plots/efficiency/pid
- plots/efficiency/dca
- plots/contamination/pid
- plots/results
- plots/results/corrected

Usage:
root -l -b -q postprocess_nuclei.C
*/
/***
This function calculates the TPC efficiency
\param[in] inFile The input file
\param[in] canvas TCanvas prepared for drawing
\param[out] fname The output pdf-file name
\param[in] pname The particle name (p, d, He4 etc)
\param[in] c_bin The centrality bin (0, 1, 2, etc)
*/
void tpc_efficiency(TFile *inFile, TCanvas *canvas, const char* fname, const char* pname, const int c_bin);
```

# Examples

Main Page Classes ▾ Files ▾

MpdNuclei
 

- ▶ MpdNuclei wagon
- ▶ Classes
- ▼ Files
  - ▶ File List
    - ▶ macros
    - MpdNuclei.cxx
    - MpdNuclei.h
    - MpdNucleiLinkDef.h
  - ▶ File Members

## macros Directory Reference

### Files

- [postprocess\\_nuclei.C](#)  
Postprocessing macro for the MpdRoot "nuclei" wagon.
- [RunAnalyses.C](#)  
Macro to run the MpdRoot analysis "train".

### ◆ tpc\_efficiency()

```
void tpc_efficiency ( TFile * inFile,
                      TCanvas * canvas,
                      const char * fname,
                      const char * pname,
                      const int c_bin
)
```

This function calculates the TPC efficiency

#### Parameters

- [in] **inFile** The input file
- [in] **canvas** TCanvas prepared for drawing
- [out] **fname** The output pdf-file name
- [in] **pname** The particle name (p, d, He4 etc)
- [in] **c\_bin** The centrality bin (0, 1, 2, etc)

Definition at line 137 of file [postprocess\\_nuclei.C](#).

## postprocess\_nuclei.C File Reference

Postprocessing macro for the MpdRoot "nuclei" wagon. [More...](#)

```
#include <stdio.h>
#include <stdlib.h>
```

[Go to the source code of this file.](#)

## Functions

```
void tpc_efficiency (TFile *inFile, TCanvas *canvas, const char *fname, const char *pname, const int c_bin)
void tof_efficiency (TFile *inFile, TCanvas *canvas, const char *fname, const char *pname, const int c_bin)
void pid_efficiency (TFile *inFile, TCanvas *canvas, const char *fname, const char *pname, const int c_bin)
void dca_efficiency (TFile *inFile, TCanvas *canvas, const char *fname, const char *pname, const int c_bin)
void pid_contamination (TFile *inFile, TCanvas *canvas, const char *fname, const char *pname, const int c_bin)
void results_pteta (TFile *inFile, TCanvas *canvas, const char *fname, const char *pname, const int c_bin)
void results_pteta_corrected (TFile *inFile, TCanvas *canvas, const char *fname, const char *pname, const int c_bin)
```

## Detailed Description

Postprocessing macro for the MpdRoot "nuclei" wagon.

New subdirectories will be created to store the produced histograms (PDF):

- plots/efficiency/tpc
- plots/efficiency/tof
- plots/efficiency/pid
- plots/efficiency/dca
- plots/contamination/pid
- plots/results
- plots/results/corrected

Usage: root -l -b -q [postprocess\\_nuclei.C](#)

Definition in file [postprocess\\_nuclei.C](#).

**Yes, that simple**  
;)