



*ROOT Users Workshop – May 10, 2022*

# **Help yourself with the proper tools to tame ROOT**

Fernando Hueso González – IFIC (CSIC/UV)



# Introduction

- Errors are a term of the equation, not an exception/accident
- There are memory leaks and thread races waiting for you
- ROOT can be particularly complicated for simple things, documentation is thus paramount!
- Equip yourself with a set of tools that optimizes these workflows
- Think of debugging as a daily tool, rather than contingency plan
- → You'll find ROOT more comfortable and user-friendly

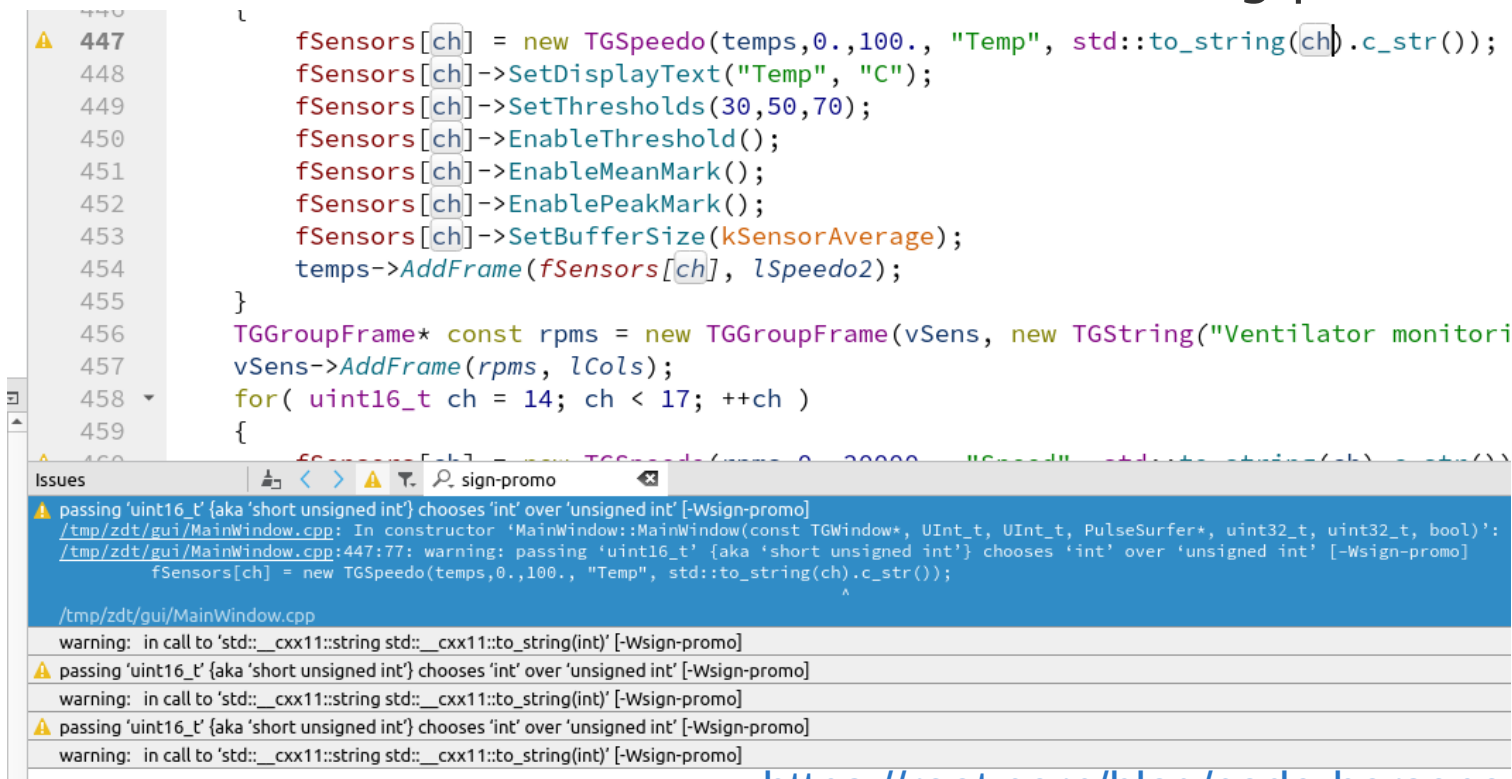
# Don't burn your (only two) eyes

- A text editor + terminal are great, but not always...

```
/tmp/zdt/gui/MainWindow.cpp: In constructor 'MainWindow::MainWindow(const TGWindow*, UInt_t, UInt_t, PulseSurfer*, uint32_t, uint32_t, bool)':
/tmp/zdt/gui/MainWindow.cpp:447:77: warning: passing 'uint16_t' {aka 'short unsigned int'} chooses 'int' over 'unsigned int' [-Wsign-promo]
    fSensors[ch] = new TGSpeedo(temps,0.,100., "Temp", std::to_string(ch).c_str());
                                     ^
/tmp/zdt/gui/MainWindow.cpp:447:77: warning: in call to 'std::__cxx11::string std::__cxx11::to_string(int)' [-Wsign-promo]
/tmp/zdt/gui/MainWindow.cpp:460:79: warning: passing 'uint16_t' {aka 'short unsigned int'} chooses 'int' over 'unsigned int' [-Wsign-promo]
    fSensors[ch] = new TGSpeedo(rpms,0.,20000., "Speed", std::to_string(ch).c_str());
                                     ^
/tmp/zdt/gui/MainWindow.cpp:460:79: warning: in call to 'std::__cxx11::string std::__cxx11::to_string(int)' [-Wsign-promo]
/tmp/zdt/gui/MainWindow.cpp:480:81: warning: passing 'uint16_t' {aka 'short unsigned int'} chooses 'int' over 'unsigned int' [-Wsign-promo]
    fSensors[ch] = new TGSpeedo(vlts,0.,12000., "Voltage", std::to_string(ch).c_str());
                                     ^
/tmp/zdt/gui/MainWindow.cpp:480:81: warning: in call to 'std::__cxx11::string std::__cxx11::to_string(int)' [-Wsign-promo]
[ 82%] Building CXX object gui/CMakeFiles/ZDTgui.dir/G_PulseSurfer.cxx.o
/tmp/zdt/gui/MainWindow.cpp: In member function 'void MainWindow::UpdateSensors()':
/tmp/zdt/gui/MainWindow.cpp:789:37: warning: '#target_mem_ref' not supported by dump_expr#<expression error>' is used uninitialized in this function [-Wuninitialized]
    float value = std::round(values[ch]);
                   ^
- - - - -
```

# Equip yourself with an IDE

- One click. No back and forth search. Warning parsed for you.



The screenshot shows a code editor with C++ code. The code defines a `TGSpeedo` object and sets its display text, thresholds, and enable flags. It then creates a `TGGroupFrame` and adds the `TGSpeedo` object to it. The code is as follows:

```
447 fSensors[ch] = new TGSpeedo(temps,0.,100., "Temp", std::to_string(ch).c_str());
448 fSensors[ch]->SetDisplayText("Temp", "C");
449 fSensors[ch]->SetThresholds(30,50,70);
450 fSensors[ch]->EnableThreshold();
451 fSensors[ch]->EnableMeanMark();
452 fSensors[ch]->EnablePeakMark();
453 fSensors[ch]->SetBufferSize(kSensorAverage);
454 temps->AddFrame(fSensors[ch], lSpeedo2);
455 }
456 TGGroupFrame* const rpms = new TGGroupFrame(vSens, new TGString("Ventilator monitori
457 vSens->AddFrame(rpms, lCols);
458 for( uint16_t ch = 14; ch < 17; ++ch )
459 {
```

Below the code, the 'Issues' panel shows a warning message:

```
passing 'uint16_t' [aka 'short unsigned int'] chooses 'int' over 'unsigned int' [-Wsign-promo]
/tmp/zdt/gui/MainWindow.cpp: In constructor 'MainWindow::MainWindow(const TGWindow*, UInt_t, UInt_t, PulseSurfer*, uint32_t, uint32_t, bool)':
/tmp/zdt/gui/MainWindow.cpp:447:77: warning: passing 'uint16_t' [aka 'short unsigned int'] chooses 'int' over 'unsigned int' [-Wsign-promo]
    fSensors[ch] = new TGSpeedo(temps,0.,100., "Temp", std::to_string(ch).c_str());
                                                                    ^
/tmp/zdt/gui/MainWindow.cpp
warning: in call to 'std::__cxx11::string std::__cxx11::to_string(int)' [-Wsign-promo]
passing 'uint16_t' [aka 'short unsigned int'] chooses 'int' over 'unsigned int' [-Wsign-promo]
warning: in call to 'std::__cxx11::string std::__cxx11::to_string(int)' [-Wsign-promo]
passing 'uint16_t' [aka 'short unsigned int'] chooses 'int' over 'unsigned int' [-Wsign-promo]
warning: in call to 'std::__cxx11::string std::__cxx11::to_string(int)' [-Wsign-promo]
```

# Equip yourself with an IDE

- Learn your compilation errors before compiling, in real-time
- Saves you tons of time!

```
445     for( uint16_t ch = 0; ch < 3; ++ch )
446     {
447         fSensors[ch] = new TGSpeedo(temps,0.,100., "Temp", std::to_string(ch).c_str());
448         fSensors[ch]->SetdisplayText("Temp", "C");
449         fSensors[ch]->SetdisplayText("30,50,70");
450         fSensors[ch]->SetdisplayText("d()");
451         fSensors[ch]->EnableMeanMark();
452         fSensors[ch]->EnablePeakMark();
453         fSensors[ch]->SetBufferSize(kSensorAverage);
454         temps->AddFrame(fSensors[ch], lSpeedo2);
455     }
456     TGroupFrame* const rpms = new TGroupFrame(vSens, new TGString("Ventilator monitoring
```

**Semantic Issue**  
448:23: error: no member named 'SetdisplayText' in 'TGSpeedo'

Annotation Settings

no member name

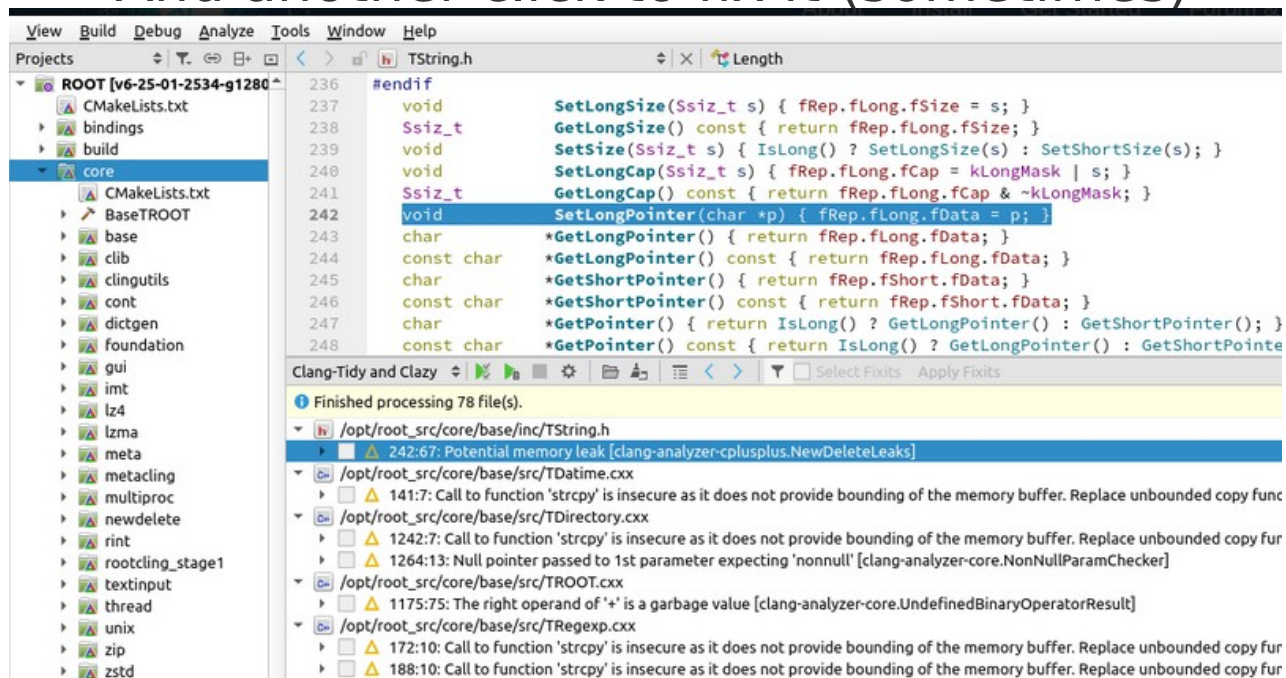
# Equip yourself with an IDE

- Get Autocomplete features while writing:

```
for( uint16_t ch = 0; ch < 3; ++ch )
{
    fSensors[ch] = new TGSpeedo(temps,0.,100., "Temp", std::to_string(ch));
    fSensors[ch]->SetDisp; // no member named 'SetDisp'
    fSensors[ch]->SetDisplayText; // void SetDisplayText(const char *text1, const char *text2 = "")
    fSensors[ch]->EnableThreshold();
    fSensors[ch]->EnableMeanMark();
    fSensors[ch]->EnablePeakMark();
    fSensors[ch]->SetBufferSize(kSensorAverage);
    temps->AddFrame(fSensors[ch], lSpeedo2);
}
```

# Equip yourself with an IDE

- Relax... and let Clang-Tidy find your bugs (static analyzer)
  - Take you with a click to the problematic place
  - And another click to fix it (sometimes)





# And if your script crashes...

- Find your segfaults, memory leaks or data races
- Stop looking at bare Helgrind / Valgrind logs...

The screenshot shows a C++ IDE with a debugger window open. The debugger window displays a list of memory issues found by Memcheck. The issues are categorized by type and location. The first issue is a 'Possible data race during read of size 1 at 0x1FFFEDEDB0 by thread #2' located in 'Application.h:148:0'. The second issue is a 'Possible data race during read of size 4 at 0xB29D098 by thread #2' located in 'TCollection.h:182:0'. The third issue is a 'Possible data race during read of size 4 at 0xB29D0AC by thread #2' located in 'TOrdCollection.h:135:0'. The fourth issue is a 'Possible data race during read of size 8 at 0x26DB7458 by thread #2' located in 'TOrdCollection.cxx:163:0'. The fifth issue is a 'Possible data race during read of size 4 at 0xB29D0B0 by thread #2' located in 'TOrdCollection.cxx:65:0'. The sixth issue is a 'Possible data race during write of size 4 at 0xB29D0AC by thread #2' located in 'TOrdCollection.cxx:70:0'. The seventh issue is a 'Possible data race during write of size 4 at 0xB29D0B0 by thread #2' located in 'TOrdCollection.cxx:84:0'.

Issue	Location
Possible data race during read of size 1 at 0x1FFFEDEDB0 by thread #2	<a href="#">Application.h:148:0</a>
Possible data race during read of size 4 at 0xB29D098 by thread #2	<a href="#">TCollection.h:182:0</a>
Possible data race during read of size 4 at 0xB29D0AC by thread #2	<a href="#">TOrdCollection.h:135:0</a>
Possible data race during read of size 8 at 0x26DB7458 by thread #2	<a href="#">TOrdCollection.cxx:163:0</a>
Possible data race during read of size 4 at 0xB29D0B0 by thread #2	<a href="#">TOrdCollection.cxx:65:0</a>
Possible data race during write of size 4 at 0xB29D0AC by thread #2	<a href="#">TOrdCollection.cxx:70:0</a>
Possible data race during write of size 4 at 0xB29D0B0 by thread #2	<a href="#">TOrdCollection.cxx:84:0</a>

- with error code 35 (EDEADLK: Resource deadlock would occur)
- 1: /usr/lib/x86\_64-linux-gnu/valgrind/vgpreload\_helgrind-amd64-linux.so
  - 2: TPoixThread::Join(TThread\*, void\*\*)
  - 3: TThread::Join(void\*\*)
  - 4: RThread::Join()
  - 5: RThread::Stop()
  - 6: MainWindow::DoStopDAQ()
  - 7: MainWindow::SaveAndExit()
  - 8: 0x19f9c029
  - 9: TClingCallFunc::exec(void\*, void\*)
  - 10: TClingCallFunc::Exec(void\*, TInterpreterValue\*)
  - 11: TCling::CallFunc\_Exec(CallFunc\_t\*, void\*) const
  - 12: TQConnection::SendSignal()
- ▼ Thread #2: Exiting thread still holds 1 lock
- 1: futex\_wait\_cancelable
  - 2: \_pthread\_cond\_wait\_common
  - 3: pthread\_cond\_wait@GLIBC\_2.3.2
  - 4: /usr/lib/x86\_64-linux-gnu/valgrind/vgpreload\_helgrind-amd64-linux.so
  - 5: TPoixCondition::Wait()
  - 6: TThread::XAResult(char const\*, int, void\*\*, int\*)
  - 7: TThread::Printf(char const\*, ...)
  - 8: RThread::Stop()
  - 9: MainWindow::DoStopDAQ()
  - 10: MainWindow::SaveAndExit()
  - 11: 0x19f9c029
  - 12: TClingCallFunc::exec(void\*, void\*)
  - 13: TClingCallFunc::Exec(void\*, TInterpreterValue\*)
  - 14: TCling::CallFunc\_Exec(CallFunc\_t\*, void\*) const

[TPoixThread.cxx:78:0](#)

[TPoixThread.cxx:78:0](#)

[TThread.cxx:528:0](#)

[RThread.cpp:89:0](#)

[RThread.cpp:113:0](#)

[MainWindow.cpp:1250:0](#)

[MainWindow.cpp:848:0](#)

[TClingCallFunc.cxx:1843:0](#)

[TClingCallFunc.cxx:2102:0](#)

[TCling.cxx:7788:0](#)

[TQConnection.h:76:0](#)

[futex-internal.h:183:0](#)

[futex-internal.h:183:0](#)

[pthread\\_cond\\_wait.c:508:0](#)

[pthread\\_cond\\_wait.c:638:0](#)

[TPoixCondition.cxx:65:0](#)

[TThread.cxx:1064:0](#)

[TThread.cxx:950:0](#)

[RThread.cpp:179:0](#)

[MainWindow.cpp:1250:0](#)

[MainWindow.cpp:848:0](#)

[TClingCallFunc.cxx:1843:0](#)

[TClingCallFunc.cxx:2102:0](#)

[TCling.cxx:7788:0](#)





# Equip yourself with an IDE

- Several options
  - QtCreator <https://root.cern/blog/code-horsepower-f1/>
  - VS-Studio <https://root.cern/blog/root-on-vscode/>
  - Eclipse  
<https://root.cern/blog/debuging-root-scripts-in-eclipse/>
- You can debug your ROOT macros (and it's worth!), not only big applications.
- You can “debug” your Doxygen documentation-build

# Help Book

## All Reference Guides

<https://root.cern/reference/>

The Reference Guide is available for all major ROOT releases. This page gives the list of all the past versions.

The screenshot shows the Qt Assistant application interface. At the top, there's a navigation bar with tabs for 'Contents', 'Index', 'Bookmarks', and 'Search'. The 'Contents' tab is active, showing a tree view of the ROOT Reference Guide. The tree is expanded to 'The Geometry Package'. The main pane displays the 'The Geometry Package' page, which includes a description of the package and a list of topics. The 'qch file' link in the top right corner is circled in red. The bottom status bar shows 'Updating search index'.

ROOT Version	HTML link	Download links	Link to the Tag file	Link to the QCH file
HEAD of the git master	<a href="#">browse</a>		<a href="#">tag file</a>	<a href="#">qch file</a>

### ROOT Reference Guide

## The Geometry Package

The ROOT geometry package is a tool for building, browsing, navigating and visualizing detector geometries. The code works standalone with respect to any tracking Monte-Carlo engine; therefore, it does not contain any constraints related to physics. However, the navigation features provided by the package are designed to optimize particle transport through complex geometries, working in correlation with simulation packages such as GEANT3, GEANT4 and FLUKA.

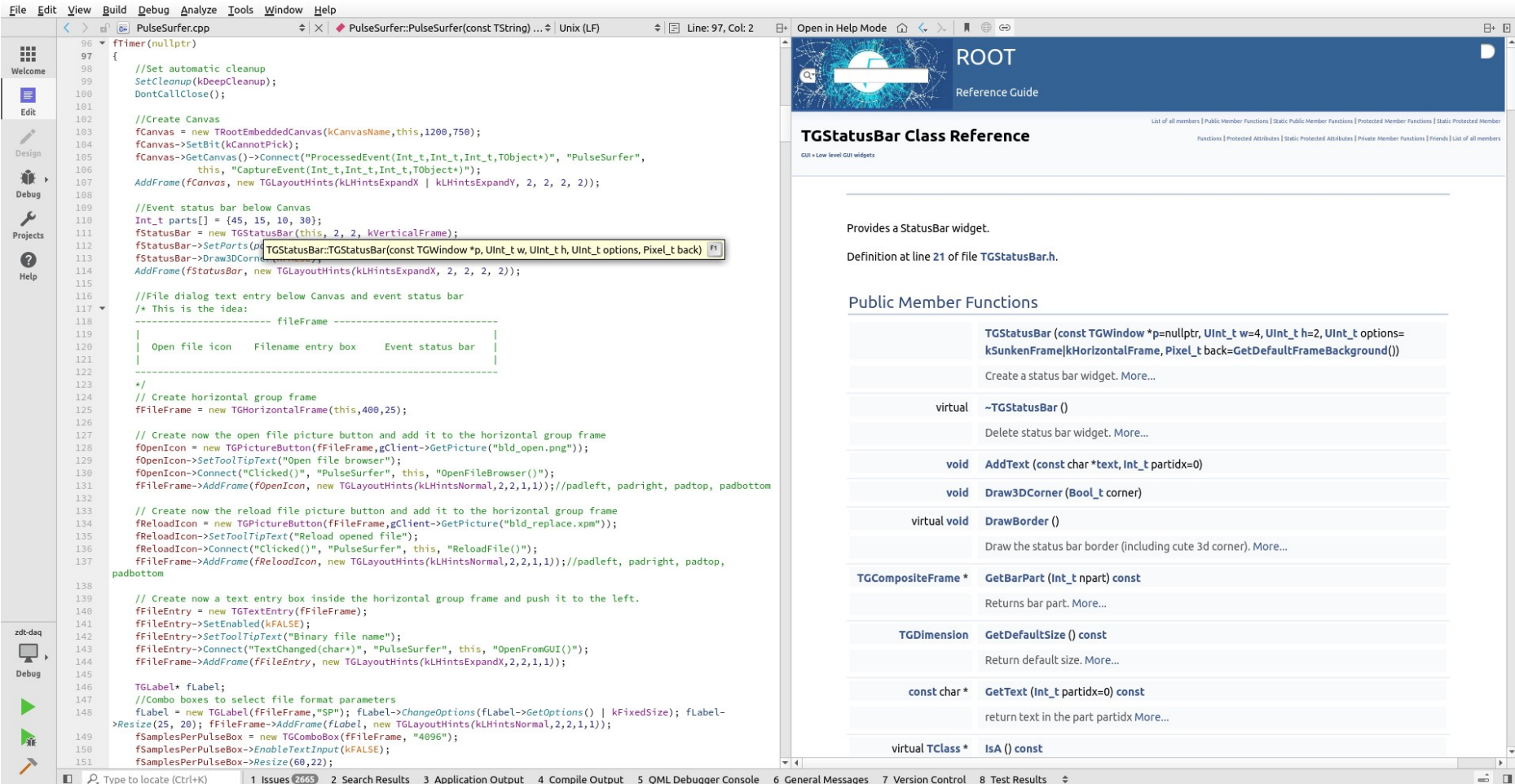
- **Quick Start: Creating the world**
  - Example 1: Creating the World
  - Example 2: A Geometrical Hierarchy Look and Feel
- **Selecting the System of Units in ROOT**
- **Geometry Creation**
  - The Volume Hierarchy

- Works offline
- Is quicker than Web

<https://root.cern/blog/code-horsepower-f1/>

qt-assistant

# The power of an F1 - QtCreator



The screenshot displays the Qt Creator IDE interface. The main editor window shows the C++ source file `PulseSurfer.cpp`. The code is a Qt widget implementation for a pulse surfer application. It includes a `timer` for periodic updates, a `Canvas` for data visualization, and a `TGStatusBar` widget for status information. The code is well-commented and includes a diagram of the GUI layout.

```
96 fTimer(nullptr)
97 {
98     //Set automatic cleanup
99     SetCleanup(kDeepCleanup);
100     DontCallClose();
101
102     //Create Canvas
103     fCanvas = new TRootEmbeddedCanvas(kCanvasName,this,1200,750);
104     fCanvas->SetBit(kCannotPick);
105     fCanvas->GetCanvas()->Connect("ProcessedEvent(Int_t,Int_t,TObject*)", "PulseSurfer",
106         this, "CaptureEvent(Int_t,Int_t,Int_t,TObject*)");
107     AddFrame(fCanvas, new TLayoutHints(kLHintsExpandX | kHintsExpandY, 2, 2, 2, 2));
108
109     //Event status bar below Canvas
110     Int_t parts[] = {45, 15, 10, 30};
111     fStatusBar = new TGStatusBar(this, 2, 2, kVerticalFrame);
112     fStatusBar->SetParts(parts, TGStatusBar::TGStatusBar(const TGWindow *p, UInt_t w, UInt_t h, UInt_t options, Pixel_t back));
113     fStatusBar->Draw3DCorner(kTopLeft, kTopRight, kBottomLeft, kBottomRight);
114     AddFrame(fStatusBar, new TLayoutHints(kLHintsExpandX, 2, 2, 2, 2));
115
116     //File dialog text entry below Canvas and event status bar
117     /* This is the idea:
118     |-----|
119     | Open file icon  Filename entry box  Event status bar |
120     |-----|
121     */
122     // Create horizontal group frame
123     fFileFrame = new TGHorizontalFrame(this,400,25);
124
125     // Create now the open file picture button and add it to the horizontal group frame
126     fOpenIcon = new TGImageButton(fFileFrame,gClient->GetPicture("bld_open.png"));
127     fOpenIcon->SetToolTipText("Open file browser");
128     fOpenIcon->Connect("Clicked()", "PulseSurfer", this, "OpenFileBrowser()");
129     fFileFrame->AddFrame(fOpenIcon, new TLayoutHints(kLHintsNormal,2,2,1,1)); //padleft, padright, padtop, padbottom
130
131     // Create now the reload file picture button and add it to the horizontal group frame
132     fReloadIcon = new TGImageButton(fFileFrame,gClient->GetPicture("bld_replace.xpm"));
133     fReloadIcon->SetToolTipText("Reload opened file");
134     fReloadIcon->Connect("Clicked()", "PulseSurfer", this, "ReloadFile()");
135     fFileFrame->AddFrame(fReloadIcon, new TLayoutHints(kLHintsNormal,2,2,1,1)); //padleft, padright, padtop,
136     padbottom
137
138     // Create now a text entry box inside the horizontal group frame and push it to the left.
139     fFileEntry = new TGTextEntry(fFileFrame);
140     fFileEntry->SetEnabled(kFALSE);
141     fFileEntry->SetToolTipText("Binary file name");
142     fFileEntry->Connect("TextChanged(char*)", "PulseSurfer", this, "OpenFromGUI()");
143     fFileFrame->AddFrame(fFileEntry, new TLayoutHints(kLHintsExpandX,2,2,1,1));
144
145     TGLabel* fLabel;
146     //Combo boxes to select file format parameters
147     fLabel = new TGLabel(fFileFrame,"SP"); fLabel->ChangeOptions(fLabel->GetOptions() | kFixedSize); fLabel-
148     >Resize(25, 20); fFileFrame->AddFrame(fLabel, new TLayoutHints(kLHintsNormal,2,2,1,1));
149     fSamplesPerPulseBox = new TGComboBox(fFileFrame, "4096");
150     fSamplesPerPulseBox->EnableTextInput(kFALSE);
151     fSamplesPerPulseBox->Resize(60,22);
```

The right sidebar shows the ROOT Reference Guide for the `TGStatusBar` class. It provides a description of the class and a list of its public member functions.

### TGStatusBar Class Reference

Provides a StatusBar widget.

Definition at line 21 of file TGStatusBar.h.

#### Public Member Functions

Function	Description
<code>TGStatusBar (const TGWindow *p=nullptr, UInt_t w=4, UInt_t h=2, UInt_t options=kSunkenFrame kHorizontalFrame, Pixel_t back=GetDefaultFrameBackground())</code>	Create a status bar widget. More...
<code>virtual ~TGStatusBar ()</code>	Delete status bar widget. More...
<code>void AddText (const char *text, Int_t partidx=0)</code>	
<code>void Draw3DCorner (Bool_t corner)</code>	
<code>virtual void DrawBorder ()</code>	Draw the status bar border (including cute 3d corner). More...
<code>TGCompositeFrame * GetBarPart (Int_t npart) const</code>	Returns bar part. More...
<code>TGDimension GetDefaultSize () const</code>	Return default size. More...
<code>const char * GetText (Int_t partidx=0) const</code>	return text in the part partidx More...
<code>virtual TClass * IsA () const</code>	

# Other useful IDE features

- Auto clang-format
- Git version control
- CTest integration

```
int main(int argc, char* argv[])  
{
```

```
int main(int argc, char *argv[]) {
```

```
[ad187b45e5] cmake/modules/RootConfiguration.cmake  
[Skipped 783 lines...] configure_file(${CMAKE_SOURCE_DIR}/cmake/scripts/  
string(REGEX REPLACE "(^| )-[M]" "" __cxxflags "${CMAKE_CXX_FLAGS}"  
string(REGEX REPLACE "(^| )-[M]" "" __cflags "${CMAKE_C_FLAGS}")  
  
if(MSVC)  
string(REPLACE "-I${CMAKE_SOURCE_DIR}/build/win" "" __cxxflags "${__c  
string(REPLACE "-I${CMAKE_SOURCE_DIR}/build/win" "" __cflags "${__cfl  
endif()  
  
if (cxxmodules)  
# Re-add the -Wno-module-import-in-extern-c which we just filtered ou  
# We want it because it changes the module cache hash and causes modu  
[Skipped unknown number of lines...]
```

bindings  
build  
ests

- ☐ Qt Test (none)
- ☐ Quick Test (none)
- ☐ Google Test (none)
- ☐ Boost Test (none)
- ☐ Catch Test (none)
- ☒ CTest
  - ☒ genetic-GAMinTutorial
  - ☒ genetic-testGAMinimizer
  - ☒ gtest-core-base-test-CoreBaseTes
  - ☒ gtest-core-base-test-CoreErrorTes
  - ☒ gtest-core-clingutils-test-corecling
  - ☒ gtest-core-cont-test-coreconttest
  - ☒ gtest-core-cont-test-testiter

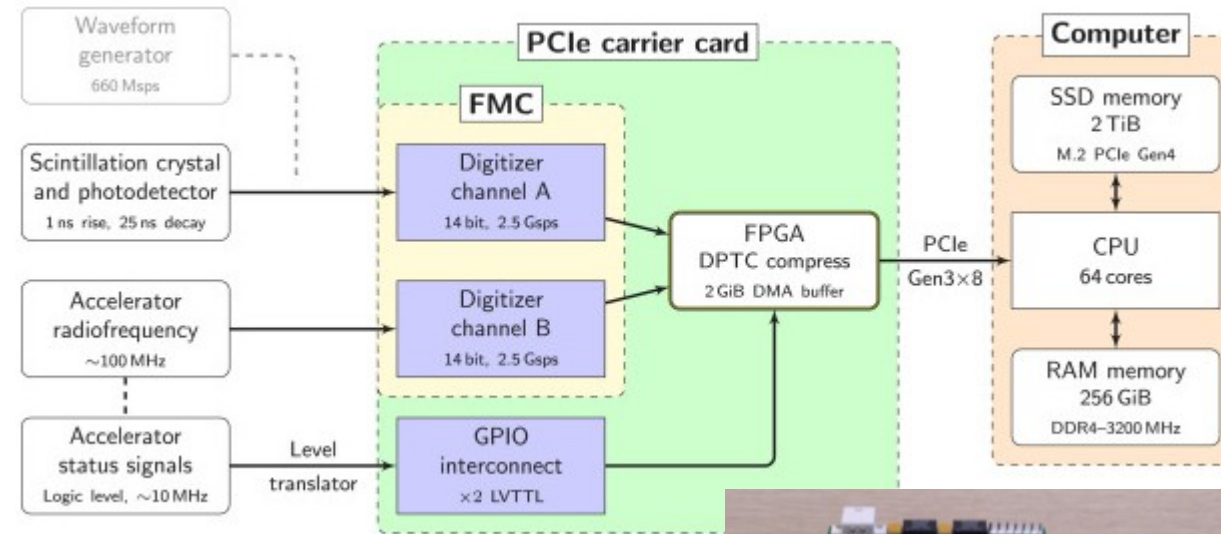
Test Results

Test summary: 43 passes, 2 fails, 2 fatals.

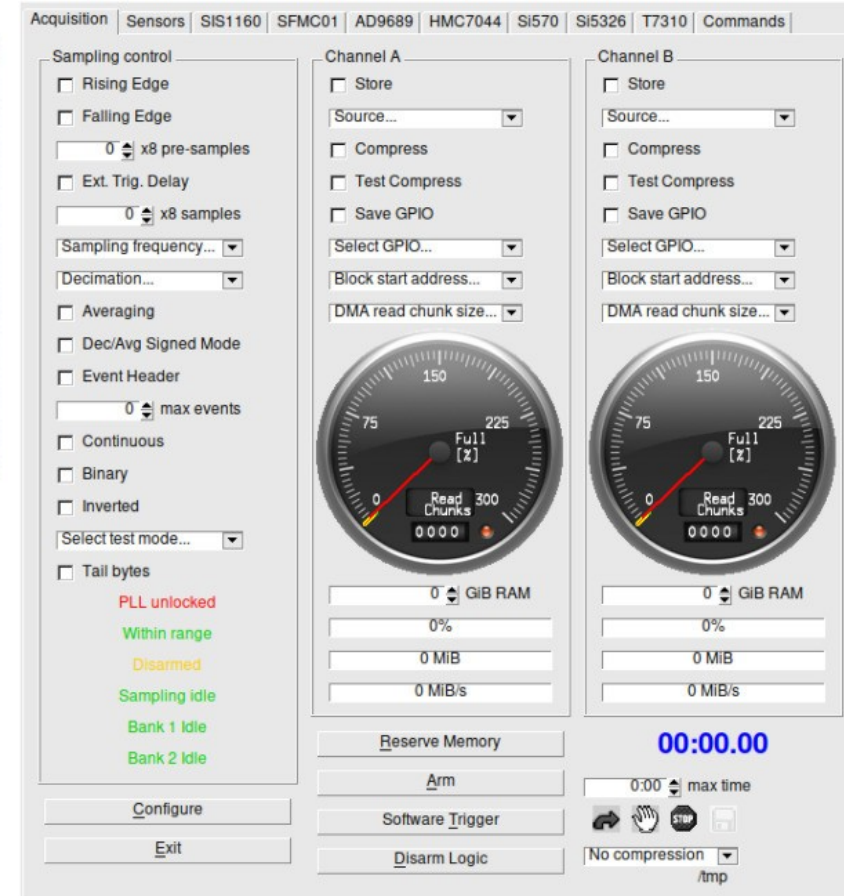
	Running tests for /home/ferhue/builds/build-root
PASS	1/810 Test #1: pyunittests-pyroot-import-lo
PASS	2/810 Test #2: pyunittests-pyroot-root-modi
FAIL	3/810 Test #3: pyunittests-pyroot-dependen
FAIL	4/810 Test #4: pyunittests-pyroot-pyz-decor
PASS	5/810 Test #5: pyunittests-pyroot-pyz-pretty
PASS	6/810 Test #7: test-import-numpy .....
PASS	7/810 Test #6: pyunittests-pyroot-pyz-array-
PASS	8/810 Test #8: pyunittests-pyroot-pyz-tobje
PASS	9/810 Test #9: pyunittests-pyroot-pyz-tobje
PASS	10/810 Test #10: pyunittests-pyroot-pyz-tclas



# Application: dead-time free DAQ



5.5 GByte/s sustained transfer speed – dead-time free data streaming for 52 seconds @ 2.5Gbps



# Proposals




- There is TH1<int>, but no TH1<long>





implement long long version of th1i th2i th3i #8546





 Open ferdymercury wants to merge 4 commits into root-project:master from ferdymercury:th1\_long 


 Conversation 62  Commits 4  Checks 4  Files changed 22

 Changes from all commits  File filter  Conversations  Settings

>   1  bindings/pyroot\_legacy/JupyROOT/helpers/cppcompleter.py 

>   2  core/rint/test/TTabComTests.cxx 

▼   2  documentation/primer/histograms.md 

 @@ -149,4 +149,4 @@ macros shows how it looks for 2D histograms:

```
149 149
150 150     ![Two 2D histograms stack on top of each other.\label{f56}][f56]
151 151
```

```
152 - [^4]: To optimise the memory usage you might go for one byte (TH1C), short (TH1S), integer (TH1I) or double-precision (TH1D) bin-content.
```

```
152 + [^4]: To optimise the memory usage you might go for one byte (TH1C), short (TH1S), integer (TH1I), long64 (TH1L64) or double-precision (TH1D) bin-content.
```



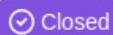
# Proposals

<https://github.com/root-project/root/issues/10474>

<https://github.com/root-project/root/issues/9491>

- Save Graph or Histograms as .txt (e.g. for later np.loadtxt)

## TGraph SaveAs .csv #9491



Closed

ferdymercury opened this issue on Jan 4 · 11 comments



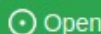
ferdymercury commented on Jan 4 · edited ▾



**Is your feature request related to a problem? Please describe.**

When working with collaborators from other disciplines, it is always problematic to share results and raw data created in ROOT in formats other than PNG or PDF.

## THistSaveAs txt or csv #10474



Open

ferdymercury opened this issue 5 days ago · 0 comments



ferdymercury commented 5 days ago · edited ▾

**Explain what you would like to see improved**

I open this to keep track of a suggestion by @lmoneta to save histograms as csv or txt files.

# Proposals

Improving Doxygen documentation, solve warnings:

- from 10k+ warnings down to 1k)
- from 4h to <1h build time

ROOT bugs	Doxygen bugs	Speedup build
<a href="https://github.com/root-project/root/pull/9655">https://github.com/root-project/root/pull/9655</a>	<a href="https://github.com/doxygen/doxygen/issues/9094">https://github.com/doxygen/doxygen/issues/9094</a>	<a href="https://github.com/root-project/root/pull/9966">https://github.com/root-project/root/pull/9966</a>
<a href="https://github.com/root-project/root/pull/9651">https://github.com/root-project/root/pull/9651</a>	<a href="https://github.com/doxygen/doxygen/issues/9144">https://github.com/doxygen/doxygen/issues/9144</a>	Migrate to CMake for easier integration

...

...


<https://github.com/root-project/web/pull/748>

# Proposals



<https://github.com/doxygen/doxygen/issues/9222>


<https://github.com/root-project/root/issues/9953>

Automated test PR for correct code documentations



 **Review required** [Show all reviewers](#)



At least 1 approving review is required by reviewers with write access. [Learn more.](#)



 **2 pending reviewers** 



 **Some checks were not successful** [Hide all checks](#)



1 failing, 2 neutral, and 2 successful checks

  **clang-tools code analysis / clang-format (pull\_request)** Failing after 41s — clang-format [Details](#)

  **LGTM analysis: Go** Completed in 2m — No code changes detected [Details](#)

  **LGTM analysis: JavaScript** Completed in 2m — No code changes detected [Details](#)

  **Jenkins CI build** — Build passed [Details](#)

  **LGTM analysis: Python** Successful in 25m — No new or fixed alerts [Details](#)

X --- Doxygen: 5 Warnings detected in changed files

[Details](#)

17 / 19

# Proposals

root [0] .gh bug



<https://github.com/root-project/root/pull/10366>

Title

Write

Preview

☐ Checked for duplicates

**Describe the bug**

**Expected behavior**

**To Reproduce**

**Setup**

```
ROOT v6.27/01
Built for linuxx8664gcc on Apr 08 2022, 16:45:48
From heads/ghissue@v6-25-01-3827-gb92902bef3
With c++ (Ubuntu 8.4.0-1ubuntu1~18.04) 8.4.0
Binary directory: /home/builds/build-root_src-Desktop_Qt_5_15_0_GCC_64bit-Debug/bin
```

**Additional context**

# Proposals



Portable parallel programming for CPU, GPU, FPGA...

<https://root-forum.cern.ch/t/root-cling-and-sycl/46014>

## ROOT, cling and SYCL

 Cling cling



ferhue

Hello dev's,

I was wondering what are the team's thought / plans with respect to SYCL. support of it within ROOT / cling envisioned? Or only later after it is shown widely accepted / used?

<https://github.com/LongyuYang/Cling-with-SYCL>

☰ README.md

## Cling with SYCL

Cling C++ Interpreter with SYCL extension. The root repository

## Installation

### Download SYCL compiler with jupyter-patch

Clone and build SYCL Compiler following [this guide](#)

## Usage

Launch cling with -fsycl argument:

```
cling -fsycl
```